



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**A SYSTEMATIC APPROACH TO DESIGN OF SPACE-  
TIME BLOCK CODED MIMO SYSTEMS**

by

Nieh, Jo-Yen

June 2006

Thesis Advisor:  
Second Reader:

Murali Tummala  
Patrick Vincent

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> June 2006	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> : A Systematic Approach to Design of Space-Time Block Coded MIMO Systems			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> :Jo-Yen Nieh				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b> <p>This thesis studies the performance of Multiple-Input, Multiple Output (MIMO) systems that use Space-Time Block Coding (STBC). Such systems can be employed to improve the bit error rate (BER) performance of wireless communication systems and counter the detrimental effects of channel fading and other distortion phenomena. We propose a systematic method for designing a space-time orthogonal MIMO scheme that employs an arbitrary number of transmitting and receiving antennas, and we evaluate (through simulation) the performance improvements that can be attained by employing our design approach. We present a general formula for determining the rate (i.e., the ratio of the number of symbols transmitted to the number of symbol intervals required) of systems that employ our design. Additionally, this thesis analyzes the relationship between channel correlation and antenna spacing for the case of MIMO systems that use a linear antenna configuration, and, through simulation studies, we show how such systems can take the advantage of the multipath phenomenon to reduce the detrimental effects of channel correlation.</p>				
<b>14. SUBJECT TERMS</b> Multiple-Input Multiple-Output (MIMO), Orthogonal Design, Space-Time Block Coding (STBC), Stanford University Interim (SUI) Models, Spatially Correlated MIMO Channels, Spatial Diversity, Alamouti Scheme, Correlation Coefficient.			<b>15. NUMBER OF PAGES</b> 123	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL	

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**A SYSTEMATIC APPROACH TO DESIGN OF SPACE-TIME BLOCK CODED  
MIMO SYSTEMS**

Jo-Yen Nieh  
Captain, Taiwan Army  
B.S., Chung Cheng Institute of Technology, 2001

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL  
June 2006**

Author: Nieh, Jo-Yen

Approved by: Murali Tummala  
Thesis Advisor

Patrick Vincent  
Second Reader

Jeffrey B. Knorr  
Chairman, Department of Electrical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

This thesis studies the performance of Multiple-Input, Multiple Output (MIMO) systems that use Space-Time Block Coding (STBC). Such systems can be employed to improve the bit error rate (BER) performance of wireless communication systems and counter the detrimental effects of channel fading and other distortion phenomena. We propose a systematic method for designing a space-time orthogonal MIMO scheme that employs an arbitrary number of transmitting and receiving antennas, and we evaluate (through simulation) the performance improvements that can be attained by employing our design approach. We present a general formula for determining the rate (i.e., the ratio of the number of symbols transmitted to the number of symbol intervals required) of systems that employ our design. Additionally, this thesis analyzes the relationship between channel correlation and antenna spacing for the case of MIMO systems that use a linear antenna configuration, and, through simulation studies, we show how such systems can take the advantage of the multipath phenomenon to reduce the detrimental effects of channel correlation.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>OBJECTIVE AND METHODOLOGY.....</b>	<b>2</b>
<b>B.</b>	<b>RELATED RESEARCH.....</b>	<b>2</b>
<b>C.</b>	<b>ORGANIZATION OF THE THESIS.....</b>	<b>3</b>
<b>II.</b>	<b>SYSTEMATIC DESIGN OF MIMO SYSTEMS .....</b>	<b>5</b>
<b>A.</b>	<b>ALAMOUTI AND HIGHER ORDER SCHEMES.....</b>	<b>5</b>
1.	2×1 Scheme .....	6
2.	2×2 Scheme .....	8
3.	3×1 Scheme .....	10
<b>B.</b>	<b>A SYSTEMATIC WAY TO DESIGN A MIMO SYSTEM OF HIGHER ORDER.....</b>	<b>12</b>
1.	Orthogonal Characteristic of H.....	12
2.	Design Process .....	13
3.	Generalized Rate Formula .....	17
4.	Example: Design of a 4×1 System.....	18
<b>C.</b>	<b>OTHER HIGH RATE CODE DESIGN .....</b>	<b>20</b>
1.	Extension of the Alamouti Scheme.....	20
2.	Rate 1/2 Code with Three Transmitting Antennas.....	23
3.	Rate 1/2 Code with Four Transmitting Antennas.....	24
<b>III.</b>	<b>CHANNEL MODEL AND ANTENNA SPACING .....</b>	<b>27</b>
<b>A.</b>	<b>THE FADING CHANNEL .....</b>	<b>27</b>
<b>B.</b>	<b>THE SUI CHANNEL .....</b>	<b>28</b>
<b>C.</b>	<b>CHANNEL CORRELATION .....</b>	<b>29</b>
<b>D.</b>	<b>ANTENNA SPACING AND CORRELATION COEFFICIENT .....</b>	<b>32</b>
1.	Laplacian Distribution.....	35
2.	Uniform Distribution .....	38
3.	Antenna Spacing .....	40
<b>IV.</b>	<b>SIMULATION RESULTS .....</b>	<b>43</b>
<b>A.</b>	<b>BASEBAND MODEL.....</b>	<b>43</b>
<b>B.</b>	<b>PERFORMANCE OF A MISO SYSTEM OVER A RAYLEIGH FADING CHANNEL.....</b>	<b>45</b>
<b>C.</b>	<b>PERFORMANCE OF A MIMO SYSTEM OVER A RAYLEIGH FADING CHANNEL.....</b>	<b>46</b>
<b>D.</b>	<b>PERFORMANCE OF A MISO SYSTEM OVER A SUI CHANNEL .....</b>	<b>49</b>
<b>E.</b>	<b>PERFORMANCE OF A MIMO SYSTEM OVER A SUI CHANNEL.....</b>	<b>51</b>
<b>F.</b>	<b>CHANNEL CORRELATION FOR THE SUI CHANNEL SCENARIO .....</b>	<b>54</b>
<b>G.</b>	<b>VARIATION IN SPACING BETWEEN ANTENNAS FOR THE SUI CHANNEL SCENARIO .....</b>	<b>55</b>
<b>V.</b>	<b>CONCLUSION .....</b>	<b>61</b>

A.	SIGNIFICANT RESULTS.....	61
B.	SUGGESTIONS FOR FUTURE WORK.....	62
APPENDIX.....		63
A.	STRUCTURE OF SIMULATION FILES .....	63
B.	EXPLANATION OF THE SUB-FUNCTIONS .....	65
C.	MATLAB CODES .....	66
LIST OF REFERENCES .....		103
INITIAL DISTRIBUTION LIST .....		105

## LIST OF FIGURES

Figure 1.	2×1 System. ....	6
Figure 2.	2×2 System. ....	8
Figure 3.	3×1 System. ....	10
Figure 4.	Flowchart of Design Process. ....	16
Figure 5.	Extension of Alamouti Scheme (After Reference [5].) ....	21
Figure 6.	Conceptual Diagram for Generating a Correlated MIMO Channel (After Reference [11].) ....	32
Figure 7.	Antenna Configuration.....	33
Figure 8.	Simplified Antenna Configuration of a Uniform Linear Array. ....	34
Figure 9.	MIMO System Configuration. ....	35
Figure 10.	Conceptual Diagram of Angle of Arrival. ....	36
Figure 11.	Magnitude of Correlation Coefficient versus Normalized Antenna Spacing for Different $\theta$ (Laplacian with $\sigma_A = 5^\circ$ , From Reference [2].) ....	37
Figure 12.	Magnitude of Correlation Coefficient versus Normalized Antenna Spacing for Different $\theta$ (Laplacian with $\sigma_A = 35^\circ$ , From Reference [2].) ....	37
Figure 13.	Magnitude of Correlation Coefficient versus Normalized Antenna Spacing (Uniform Distribution from $-\pi$ to $\pi$ , From Reference [2].) ....	40
Figure 14.	Magnitude of Correlation Coefficient versus Antenna Spacing for Different $\theta$ (Laplacian with $\sigma_A = 5^\circ$ , After Reference [2].) ....	41
Figure 15.	Magnitude of Correlation Coefficient versus Antenna Spacing for Different $\theta$ (Laplacian with $\sigma_A = 35^\circ$ , After Reference [2].) ....	41
Figure 16.	QPSK Constellation Plot. ....	44
Figure 17.	BER Performance of MISO Systems over a Rayleigh Fading Channel. ....	46
Figure 18.	BER Performance of MIMO Systems over a Rayleigh Fading Channel (Two Transmitting Antennas.) ....	47
Figure 19.	BER Performance of MIMO Systems over a Rayleigh Fading Channel (Three Transmitting Antennas.) ....	48
Figure 20.	BER Performance of MIMO Systems over a Rayleigh Fading Channel (Four Transmitting Antennas.) ....	48
Figure 21.	BER Performance of MISO Systems over a SUI Channel (Directional Antennas.) ....	50
Figure 22.	BER Performance of MISO Systems over a SUI Channel (Omni-Antennas.) ....	51
Figure 23.	BER Performance of Two-Transmitter MIMO Systems over a SUI Channel (Two Directional Transmitting Antennas.) ....	52
Figure 24.	BER Performance of Three-Transmitter MIMO Systems over a SUI Channel (Three Directional Transmitting Antennas.) ....	53
Figure 25.	BER Performance of Four-Transmitter MIMO Systems over a SUI Channel (Four Directional Transmitting Antennas.) ....	53

Figure 26.	BER Performance Comparison of a 4×1 MISO System over a SUI-2 Channel with Correlation Coefficient Variation (Four Directional Antennas.) .....	54
Figure 27.	Truncated Curve of the Magnitude of the Correlation Coefficient versus Antenna Spacing ( $\theta = 0^0$ , Laplacian AOA with $\sigma_A = 5^\circ$ ). .....	57
Figure 28.	Performance with Spacing Variation (2×1 system.) .....	58
Figure 29.	Performance with Spacing Variation (3×1 system.) .....	58
Figure 30.	Performance with Spacing Variation (4×1 system.) .....	59
Figure 31.	A Conceptual Diagram of Simulation Structure (After Reference [11]) .....	64

## LIST OF TABLES

Table 1.	SUI Channel Models and Their Characterization (From Reference [11].).....	29
Table 2.	Simulation of a MISO System over a Rayleigh Fading Channel. ....	45
Table 3.	Required $E_b/N_0$ for a BER of $10^{-6}$ for Different MIMO Schemes .....	49
Table 4.	Parameters of a MISO System over a SUI Channel. ....	50

THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

I thank my beloved wife Wen-Tzu for her endless encouragement and abundant support during the period of studying at NPS. With her help, we had a wonderful memory in Monterey.

I would like to thank my advisor Murali Tummala for his patiently tutoring during the whole process of thesis. He is very knowledgeable in several fields and hopefully my thesis can help his research in some sense. I would like to thank my second reader Patrick Vincent for his help in providing many important suggestions.

Several friends are needed to be addressed here for remembering the friendship during this wonderful studying period. They are Mr. Kao, Chi-Han, Mr. Yang, Jia-Hong, and Mr. Tsai, Wen-Hsiang. Thank all of you for every heart-warmed gathering and happy moments.

Last but not the least, I deeply thank the Lord for giving me life and the opportunity to enjoy the life.

THIS PAGE INTENTIONALLY LEFT BLANK



## EXECUTIVE SUMMARY

Traditionally, wireless communication entails a sender using a single antenna to transmit a signal, which after undergoing modification in the communication channel, is then received by a single antenna at the receiver. In a Multiple-Input, Multiple-Output (MIMO) scheme, on the other hand, the sender uses multiple antennas for transmission, and the receiver uses multiple antennas for reception, thus making available multiple communication paths (i.e., channels) between the sender and receiver. These multiple channels can be used to increase the data rate by sending different data streams on the different channels. Alternatively, by transmitting a block of carefully encoded data on each of the different channels—a procedure known as Space Time Block Coding (STBC)—the bit error rate (BER) performance at the receiver can be improved.

In this thesis, we focus on the employment of MIMO systems that use STBC to improve the BER performance of wireless communication systems in fading channels, as well as in channels based on actual field measurements. In a fading channel, the transmitted signal is reflected, diffracted or scattered in the channel, and, as a consequence, attenuated “versions” of the signal arrive at the receiver via multiple paths and at varying times, causing unpredictable time-varying changes in the magnitude and phase of the arriving received signal. Real-world channels will also exhibit Doppler spread and co-channel interference as well.

A variety of different MIMO schemes were evaluated in simulation using the MATLAB programming language. We show that MIMO systems can be used to improve the BER performance of wireless communication systems and can counter the effects of channel fading and other distortion phenomena. We propose a systematic process for designing a MIMO system with an arbitrary number of transmitting antennas and receiving antennas, and evaluate (through simulation) the performance improvements that can be attained by employing our design, in both Rayleigh fading channels and in channels based on actual measurements (the Stanford University Interim channel model). We present a general formula that can be readily used to determine the rate (i.e., the ratio of the number of symbols transmitted over the number of symbol intervals required) of

systems that employ our design. Three criteria—the rate of the scheme, the BER and the complexity—are used to analyze our scheme against alternative MIMO designs available in the literature, and we describe the trade-offs that a designer must consider when contemplating the choice of a MIMO system in a specific application.

In addition to fading, the correlation between channels may also cause significant degradation in the performance of communication systems. The degree of correlation is a function of the probability distribution of the angle of arrival of the incoming signal, as well as the antenna spacing, antenna configuration and wavelength. In this thesis, we study the relationship between the antenna spacing and the channel correlation for the case of a uniform linear antenna configuration. Through analysis and simulation, we analyze the effects of channel correlation on the performance of MIMO systems. We note that MIMO systems can employ the multipath phenomenon to reduce the detrimental effects of channel correlation; the greater the number of multi-path arrivals, the less correlated the channels will be. Thus, a MIMO system can take advantage of the large number of multi-path arrivals to achieve better performance.

# I. INTRODUCTION

Traditionally, wireless communication entails a sender using a single antenna to transmit a signal, which, after undergoing modification in the communication channel, is then received by a single antenna at the receiver. In a Multiple-input, Multiple-output (MIMO) scheme, on the other hand, the sender uses multiple antennas for transmission, and the receiver uses multiple antennas for reception, thus making available multiple communication paths (i.e., channels) between the sender and receiver. These multiple channels can be used to increase the data rate by sending different data streams on the different channels. Alternatively, by sending the same signal on each of the different channels, the bit error rate (BER) performance at the receiver can be improved.

In this thesis, we focus on the use of MIMO systems to improve the BER performance of wireless communication systems in fading channels as well as in channels based on actual field measurements. In a fading channel, the transmitted signal is reflected, diffracted or scattered in the channel, and, as a consequence, attenuated “versions” of the symbol arrive at the receiver via multiple paths and at varying times, causing unpredictable time-varying changes in the magnitude and phase of the arriving received signal. Real-world channels will also exhibit Doppler spread and co-channel interference as well.

This thesis studies the performance of a variety of MIMO schemes in channels exhibiting fading and other distortion phenomena. We show that by properly designing the system to meet orthogonality conditions, a system employing multiple antennas can achieve a markedly improved bit error rate. We propose a systematic process for designing a MIMO system with an arbitrary number of transmitting antennas and receiving antennas, and evaluate (through simulation) the performance improvements that can be attained by employing our design approach.

In addition to fading, the correlation between channels may cause significant degradation in the performance of communication systems. As shown in [1] and [2], channel correlation is a function of probability distribution of the angle of arrival of the incoming signal, as well as the antenna spacing, antenna configuration, and wavelength.

In this thesis, we study the relationship between the antenna spacing and the channel correlation for the case of a linear array antenna configuration. Through simulation studies, how MIMO systems can counter the effects of channel correlation is discussed in this thesis. The simulation results manifest that the multi-path effect helps the reduction of antenna correlations.

The verification of a space-time coded orthogonal MIMO scheme over a fading channel is a big part of this thesis. By properly designing the orthogonality of a wireless channel and transmitting signal, a system employing multiple antennas can achieve lower symbol error rate and better performance. From the references of published works, a systematic designing process of a MIMO scheme is developed and authenticated in the simulation. The results indicate the improvements of a MIMO scheme.

#### **A. OBJECTIVE AND METHODOLOGY**

The goal of this thesis is to investigate a MIMO system and propose a method for designing a space-time orthogonal MIMO scheme with simulation verification. Additionally, we analyze the relationship between channel correlation coefficients and the antenna spacing for the case of linear antenna configurations. Our results make use of the published results in the literature that apply to formulas for different combinations of antennas and their configurations. The space-time coding schemes are investigated and summarized in terms of performance, methods of signal recovery, diversity gain, and communication rate. A widely-used channel model is adopted for simulation purposes.

#### **B. RELATED RESEARCH**

The Alamouti's transmit diversity scheme [3] laid the foundation for the orthogonal design of MIMO systems. Since then, numerous designs have been proposed that offered improved performance for higher diversity and higher communication rates. Since these schemes employ multiple channels (space diversity) and careful consideration of what symbol should be sent at any specific time (time diversity), the method for encoding and decoding data is termed "Space-Time Block Coding (STBC)." The mathematical frame work of orthogonal design of such codes is discussed in [4] further. Some variations of STBC codes are presented in [5]. These different orthogonal designs provide orthogonal characteristics and satisfactory fading resistance. Reference [6] summarizes the recent work performed as of 2004.

Channel correlation is another active area of research. Formulas for correlation coefficients when the probability distribution of the angle of arrival of the incoming signal is Laplacian and uniform are presented in [1] and [2]. The type of distribution significantly influences the correlation coefficient values. The case in which the angle of arrival follows a Gaussian distribution was studied in [7].

### **C. ORGANIZATION OF THE THESIS**

This thesis is composed of five chapters. Chapter II introduces the fundamentals of MIMO systems that employ several antenna combinations with STBC. We propose a method for designing a space-time orthogonal MIMO scheme that employs an arbitrary number of transmitting and receiving antennas, and we present a generalized formula for the rate of our design. We present an example illustrating the designing of a system with four transmit antennas and one receive antenna, and then explain how the design is readily extended to a system with four transmit antennas and an arbitrary number of receive antennas. Chapter III introduces the nature of a fading channel and the Stanford University Interim (SUI) channel, which is based on field measurements [8]. The computation of correlation coefficients between channels is discussed and exhibited, and the relationship between antenna spacing and normalized wavelength for a given signal distribution model and broadside angle is manifested. Chapter IV presents the simulation results showing the BER performance of each communication scheme in a variety of channels. Chapter V summarizes our conclusions and suggestions for future research work. Appendix describes the Matlab functions used to generate the results in Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. SYSTEMATIC DESIGN OF MIMO SYSTEMS

This chapter introduces Multiple-Input, Multiple-Output (MIMO) systems and describes their characteristics. In addition, a systematic method for designing MIMO systems is proposed. MIMO systems have attracted considerable research attention since they afford an improvement in communication performance without requiring a corresponding increase in communication bandwidth.

Typically, classical single-input, single-output (SISO) wireless communication entails a sender using a single antenna to transmit a signal, which is subsequently modified by the communication channel and received by a single antenna at the receiver. In a MIMO scheme, on the other hand, the sender uses multiple antennas for transmission, and the receiver uses multiple antennas for reception, thus making available multiple communication paths (i.e., channels) between the sender and receiver. These multiple channels can be used to increase the data rate by sending different data streams on the different channels. Alternatively, by sending the same signal on each of the different channels, the bit error performance at the receiver can be improved. The number of independent channels that a signal traverses from the sender to the receiver is termed the diversity gain. [6]. The maximum achievable diversity gain is the product of the number of transmitting antennas and the number of receiving antennas, with the assumption that all signal paths are independent.

Correct operation of MIMO systems requires careful design, with the encoded signals from each transmitting antenna and/or the multiple communication channels meeting specified orthogonality conditions. In this chapter, a systematic method for designing MIMO systems that satisfy such orthogonality conditions will be presented.

### A. ALAMOUTI AND HIGHER ORDER SCHEMES

A simple and effective MIMO system using two antennas at the transmitting side and one or more antennas at the receiving side was first presented by Alamouti [3]. In this section, we review the basic concepts of the Alamouti scheme, and then use these fundamental concepts to introduce higher order schemes (that employ more than two

antennas at the transmitting and/or receiving ends) and achieve different communication rates. The correctness of our design is verified by simulation.

### 1. 2×1 Scheme

This scheme uses two antennas for transmitting and one for receiving as shown in Figure 1. The diversity gain is double that of the SISO system. The scheme operates by transmitting two symbols,  $x_1$  and  $x_2$ , during two time intervals as follows: the symbols  $x_1$  and  $x_2$  are simultaneously transmitted during the first time interval using antennas 1 and 2, respectively, and then the symbols  $-x_2^*$  and  $x_1^*$  are simultaneously transmitted during the second time interval. Thus, the transmitted signal expressed as a matrix is

$$X = \begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix}$$

where the asterisk  $*$  indicates complex conjugation operation. This process is then repeated to transmit the next two symbols, and so forth. Figure 1 shows how the four symbols  $x_1, x_2, x_3$  and  $x_4$  would be transmitted during four consecutive time intervals. This scheme has the characteristic that two symbols are simultaneously encoded and decoded as a block, a point that will be demonstrated in what follows.

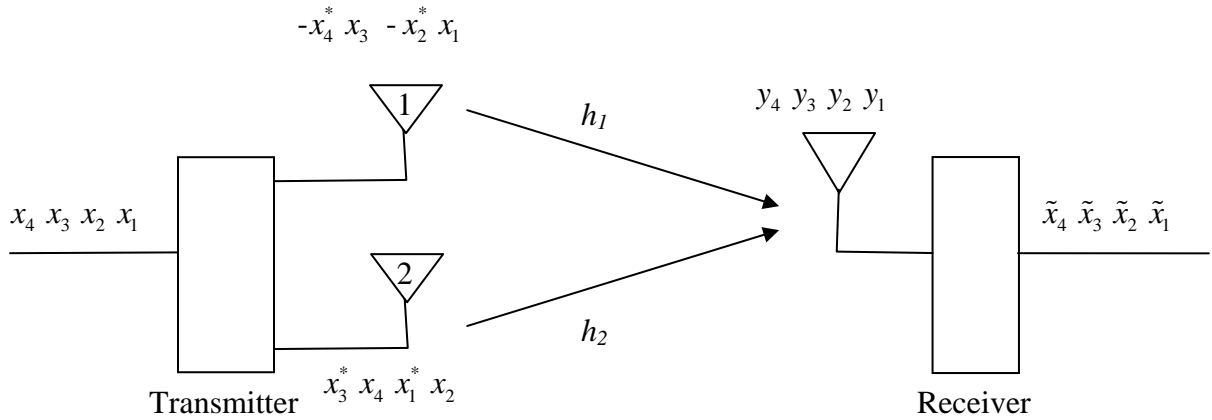


Figure 1. 2×1 System.

Referring to Figure 1, the received signals  $y_1$  and  $y_2$  are the combination of the signals from antennas 1 and 2, plus additive noise:



$$y_1 = h_1 x_1 + h_2 x_2 + n_1 \quad (2.1)$$

$$y_2 = -h_1^* x_2 + h_2^* x_1 + n_2$$

where  $n_1$  and  $n_2$  represent the additive noise and  $h_1$  and  $h_2$  are complex numbers representing the channel gain. We assume that the channel is memoryless and flat, thus inducing a magnitude and phase change in the transmitted signal, without inducing a time delay. We further assume that the receiver has perfect Channel State Information (CSI); i.e., the receiver can measure and determine  $h_1$  and  $h_2$ .

The receiver then computes estimates of  $x_1$  and  $x_2$ , denoted  $\tilde{x}_1$  and  $\tilde{x}_2$ , by processing  $y_1$  and  $y_2$  as follows [3]:

$$\begin{aligned} \tilde{x}_1 &= h_1^* y_1 + h_2^* y_2 \\ \tilde{x}_2 &= h_2^* y_1 - h_1^* y_2 \end{aligned} \quad (2.2)$$

Substituting (2.1) into (2.2) we obtain

$$\begin{aligned} \tilde{x}_1 &= h_1^* y_1 + h_2^* y_2 = (|h_1|^2 + |h_2|^2) x_1 + h_1^* n_1 + h_2^* n_2 \\ \tilde{x}_2 &= h_2^* y_1 - h_1^* y_2 = (|h_1|^2 + |h_2|^2) x_2 + h_2^* n_1 - h_1^* n_2 \end{aligned} \quad (2.3)$$

Thus, the estimated  $\tilde{x}_1$  and  $\tilde{x}_2$  are proportional to the values of  $x_1$  and  $x_2$  (in the presence of additive noise terms), and any phase variations that might occur in the channel have no effect on the estimated signals. Additionally, any fading in either channel will affect the two estimates in the same way, and the probability that both independent channels will fade simultaneously is lower than that of only a single channel being subjected to fading. The Alamouti scheme provides an effective way to combat phase and magnitude changes that occur in a fading channel, and thus provides a means for reducing the bit error rate (BER).

Using matrix notation, (2.1) can be rewritten as

$$\begin{bmatrix} y_1 \\ y_2^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.4)$$

where  $H$  is the channel matrix given by

$$H = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \quad (2.5)$$

A matrix such as  $H$  is said to have an *orthogonal design* [6]. Using (2.8), we can see that

$$H^H H = \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} = (|h_1|^2 + |h_2|^2) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.6)$$

where  $H^H$  is the Hermetian (i.e., the complex-conjugated transpose) of  $H$ . From the definition of orthogonality, a simple way to determine if a matrix has an orthogonal design is to check if each column is orthogonal with the complex-conjugate of all other columns. For example, we can see that the Alamouti scheme exhibits orthogonal design by multiplying the columns as following: is  $h_1 \times h_2^* - h_2^* \times h_1 = 0$ . All the schemes presented in this thesis will exhibit this orthogonal design characteristic.

The Alamouti scheme uses two symbol intervals to transmit two symbols, and thus has a rate of one. In the following sections, we shall see that as the number of transmitting antennas increases, the rate decreases. The two-by-two channel matrix  $H$  can be interpreted as two time intervals (time diversity) by two antennas (spatial diversity). A scheme incorporating multiple antennas and orthogonal design is called a Space-Time Block Coding (STBC) scheme.

## 2. 2×2 Scheme

The 2×2 scheme is an extension of the 2×1 scheme. Figure 2 illustrates the system.

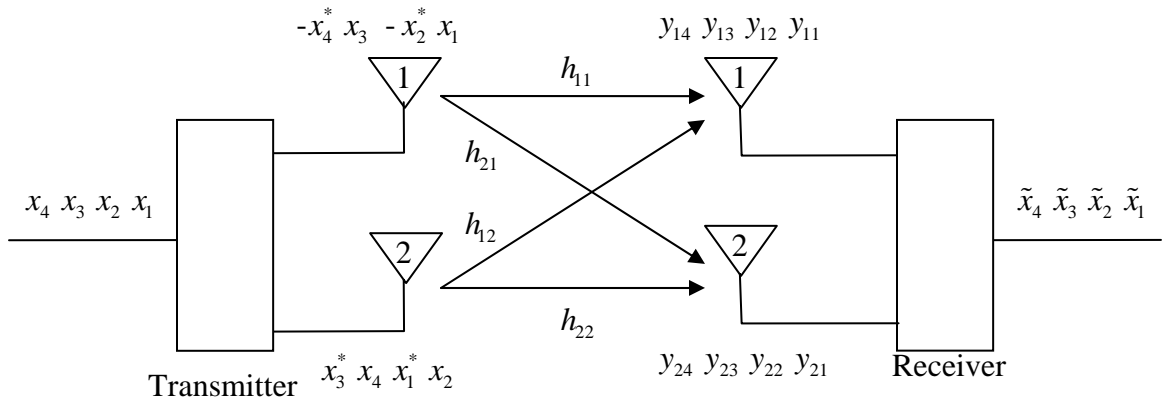


Figure 2. 2×2 System.

The diversity gain is double that of the  $2 \times 1$  scheme since there are four different paths. The transmitting side is the same as the  $2 \times 1$  scheme. Antenna 1 transmits  $x_1$  and antenna 2 transmits  $x_2$  during the first time interval. In the second time interval,  $-x_2^*$  and  $x_1^*$  are transmitted by antennas 1 and 2, respectively. Two antennas are on the receiver side. Focusing on antenna 1 on the receiving side, its first reception,  $y_{11}$ , is the combination of two signals transmitted during the first time interval, and its second reception,  $y_{12}$ , is the combination of the two signals transmitted during the second time interval, and so on. The received signals can be written as

$$\begin{aligned} y_{11} &= h_{11}x_1 + h_{12}x_2 + n_1 \\ y_{12} &= -h_{11}x_2^* + h_{12}x_1^* + n_2 \\ y_{21} &= h_{21}x_1 + h_{22}x_2 + n_3 \\ y_{22} &= -h_{21}x_2^* + h_{22}x_1^* + n_4 \end{aligned} \quad (2.7)$$

or in matrix form of

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix} + \begin{bmatrix} n_1 & n_2 \\ n_3 & n_4 \end{bmatrix}$$

The receiver uses a similar relation to decode and estimate the transmitted signals as that used in the  $2 \times 1$  scheme. The symbols  $x_1$  and  $x_2$  are reconstructed as

$$\begin{aligned} \tilde{x}_1 &= h_{11}^*y_{11} + h_{12}y_{12}^* + h_{21}^*y_{21} + h_{22}y_{22}^* \\ \tilde{x}_2 &= h_{12}^*y_{11} - h_{11}y_{12}^* + h_{22}^*y_{21} - h_{21}y_{22}^* \end{aligned} \quad (2.8)$$

By substituting (2.7) into (2.8), we have the recovered symbols

$$\begin{aligned} \tilde{x}_1 &= (|h_{11}|^2 + |h_{12}|^2 + |h_{21}|^2 + |h_{22}|^2)x_1 + \eta_1 \\ \tilde{x}_2 &= (|h_{11}|^2 + |h_{12}|^2 + |h_{21}|^2 + |h_{22}|^2)x_2 + \eta_2 \end{aligned} \quad (2.9)$$

where the noise terms  $\eta_1$  and  $\eta_2$  are given by

$$\begin{aligned} \eta_1 &= h_{11}^*n_1 + h_{12}n_2^* + h_{21}^*n_3 + h_{22}n_4^* \\ \eta_2 &= h_{12}^*n_1 - h_{11}n_2^* + h_{22}^*n_3 - h_{21}n_4^* \end{aligned}$$

Comparing Equations (2.7) to (2.8) with Equations (2.1) and (2.2), it can be observed that the  $2 \times 2$  scheme is the combination of two  $2 \times 1$  schemes, assuming there is no correlation between the two antennas on both the transmitting and receiving sides. The diversity gain and the magnitude of estimated signals will linearly increase as the number

of receiving antennas increases, but since the number of antennas on the transmitting side does not change, the rate will remain fixed.

More importantly, these conclusions can be extended to any  $2 \times r$  scheme as observed in [3], where  $r$  is the number of receiving antennas. Due to the increment in the magnitude of the estimated signal, the BER is reduced significantly as the number of receiving antennas increases from one to two. This result will be demonstrated in Chapter IV.

### 3. 3×1 Scheme

A  $3 \times 1$  scheme is presented in [6]. This scheme provides a transmitting matrix indicating which symbol should be transmitted by each antenna during any specific symbol interval. The system is illustrated in Figure 3.

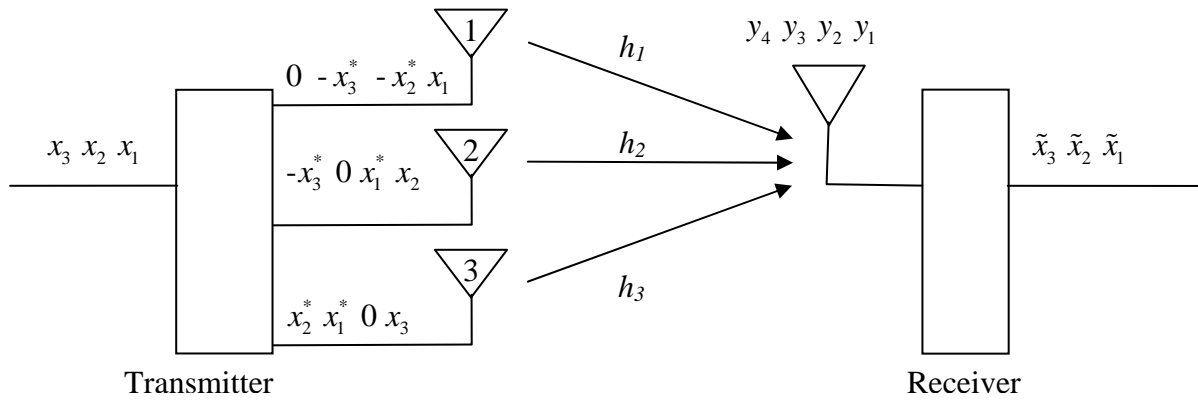


Figure 3. 3×1 System.

The transmitting matrix  $X$  is

$$X = \begin{bmatrix} x_1 & -x_2^* & -x_3^* & 0 \\ x_2 & x_1^* & 0 & -x_3^* \\ x_3 & 0 & x_1^* & x_2^* \end{bmatrix}$$

which shows that four symbol intervals are needed in order to transmit three symbols.

From Figure 3, the received signals can be determined as

$$y_1 = h_1 x_1 + h_2 x_2 + h_3 x_3 + n_1$$

$$y_2^* = -h_1^* x_2 + h_2^* x_1 + n_2^*$$

$$y_3^* = -h_1^* x_3 + h_3^* x_1 + n_3^*$$

$$y_4^* = -h_2^* x_3 + h_3^* x_2 + n_4^*$$

which can be rewritten in matrix notation with the noise terms ignored here for convenience as

$$\underline{y} = H\underline{x} + \underline{n} \tag{2.10}$$

$$\begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_2^* & -h_1^* & 0 \\ h_3^* & 0 & -h_1^* \\ 0 & h_3^* & -h_2^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3^* \\ n_4^* \end{bmatrix}$$

where  $H$  is the channel matrix. It is easy to check that the matrix exhibits the property of orthogonality by computing

$$H^H H = (|h_1|^2 + |h_2|^2 + |h_3|^2) I_3$$

where  $I_3$  is a  $3 \times 3$  identity matrix. Since four time intervals are required to transmit three symbols, the rate of the  $3 \times 1$  scheme is  $3/4$ . The diversity gain is greater than that for the  $2 \times 1$  scheme, assuming non-correlated channels. The order of channel matrix  $H$  is  $4 \times 3$ , which can be interpreted as four time intervals (time diversity) by three antennas (spatial diversity), similar to the interpretation used in the  $2 \times 1$  scheme.

The estimated symbols vector is given by

$$\underline{\tilde{x}} = H^H \underline{y}$$

which can be expanded into

$$\tilde{x}_1 = h_1^* y_1 + h_2 y_2^* + h_3 y_3^*$$

$$\tilde{x}_2 = h_2^* y_1 - h_1 y_2^* + h_3 y_4^*$$

$$\tilde{x}_3 = h_3^* y_1 - h_1 y_3^* - h_2 y_4^*$$

and, by substituting (2.10) into above, it can be shown that

$$\tilde{x}_1 = \left( |h_1|^2 + |h_2|^2 + |h_3|^2 \right) x_1 + \eta_1$$

$$\tilde{x}_2 = \left( |h_1|^2 + |h_2|^2 + |h_3|^2 \right) x_2 + \eta_2$$

$$\tilde{x}_3 = \left( |h_1|^2 + |h_2|^2 + |h_3|^2 \right) x_3 + \eta_3$$

where the noise terms are given by

$$\eta_1 = h_1^* n_1 + h_2 n_2^* + h_3 n_3^*$$

$$\eta_2 = h_2^* n_1 - h_1 n_2^* + h_3 n_4^*$$

$$\eta_3 = h_3^* n_1 - h_1 n_3^* - h_2 n_4^*$$

Since the magnitude of estimated signals is now greater, it is reasonable to conclude that the BER performance will be improved over the 2×1 system. The results presented for the diversity gain, rate, and recovered signals can be extended to 3×M systems by using the same steps shown in the previous section.

## **B. A SYSTEMATIC WAY TO DESIGN A MIMO SYSTEM OF HIGHER ORDER**

This section proposes a systematic method for designing an arbitrary Multiple-Input, Single-Output (MISO) system, and then extends the design to a MIMO system with an arbitrary number of receiving antennas. In other words, this method can be used to design a system with any combination of antenna sets, and simulation results will demonstrate the improvement as the order of the MIMO system increases. The key point of this design is ensuring the orthogonality of the channel matrix.

### **1. Orthogonal Characteristic of H**

The orthogonality property is verified by either computing  $H^H H$  and showing that the result is a diagonal matrix, or by ensuring that each column is orthogonal to all the other columns. These two tests will be used repeatedly in the following design

process. Also note that the number of rows in the channel matrix is equal to the time diversity, and the number of columns is equal to the space diversity.

## 2. Design Process

Consider that a MIMO system with  $t$  transmitting antennas and  $r$  receiving antennas is to be designed. A MISO system with  $t$  transmitting antennas is first developed, and then expanded to a MIMO system with  $r$  receiving antennas.

Step 1: In formulating the channel matrix, we begin with the first row containing  $t$  columns (recall that  $t$  equals the spatial diversity) as

$$\begin{bmatrix} h_1 & h_2 & h_3 & \dots & h_t \end{bmatrix}$$

The objective is then to obtain additional rows in such a manner that each column in the final matrix is orthogonal to all others. We fill in the second row, with the purpose of making the first and second columns orthogonal, as

$$\begin{bmatrix} h_1 & h_2 & h_3 & \dots & h_t \\ -h_2^* & h_1^* & 0 & \dots & 0 \end{bmatrix}$$

Now, we add a third row to make the first and third columns orthogonal, then a fourth row to make the first and fourth columns orthogonal and so on. In total, we add  $t-1$  rows to make the first column orthogonal to all other columns. Then, we turn our attention to the second column. We add a row to make the second and third columns orthogonal, then another row to make the second and fourth columns orthogonal, and so forth. A total of  $t-2$  rows will be needed to make the second column orthogonal to all others. We continue this process, until we arrive at the point where only a single row must be added to make the  $(t-1)^{th}$  column orthogonal to the  $t^{th}$  column. A matrix so chosen is of the form

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots & h_{t-1} & h_t \\ h_2^* & -h_1^* & 0 & 0 & \dots & 0 & 0 \\ h_3^* & 0 & -h_1^* & 0 & \dots & 0 & 0 \\ h_4^* & 0 & 0 & -h_1^* & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & h_t^* & -h_{t-1}^* \end{bmatrix} \quad (2.11)$$

This fulfills the orthogonality requirement for the channel matrix. The number of rows indicates the number of symbol intervals needed to transmit  $t$  symbols. As noted, we start with the top row, then add  $j-1$  rows to make the first column orthogonal to columns 2 through  $j$ , then add  $j-2$  rows to make the second column orthogonal to columns 3 through  $j$ , and so forth. Thus, the total number of rows in the completed channel matrix is

$$1 + \sum_{j=1}^{t-1} j = 1 + \frac{t(t-1)}{2} = \frac{t^2 - t + 2}{2}$$

Step 2: The input-output relationship using the designed  $H$  matrix is thus given by

$$\underline{y} = H\underline{x} + \underline{n}$$

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4^* \\ y_5^* \\ \vdots \\ y_{(t^2-t+2)/2}^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots & h_{t-1} & h_t \\ -h_2^* & h_1^* & 0 & 0 & \dots & 0 & 0 \\ -h_3^* & 0 & h_1^* & 0 & \dots & 0 & 0 \\ -h_4^* & 0 & 0 & h_1^* & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -h_t^* & h_{t-1}^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3^* \\ n_4^* \\ n_5^* \\ \vdots \\ n_{(t^2-t+2)/2}^* \end{bmatrix} \quad (2.12)$$

where  $\underline{y}$  is a  $(\frac{t^2-t+2}{2}) \times 1$  vector of received signals,  $H$  is a  $(\frac{t^2-t+2}{2}) \times t$  channel matrix,  $\underline{x}$  is a  $t \times 1$  vector of input symbols, and  $\underline{n}$  is a  $(\frac{t^2-t+2}{2}) \times 1$  vector of noise. The complex conjugation terms in  $\underline{y}$  from  $y_2^*$  to  $y_{(t^2-t+2)/2}^*$  are due to the nature of this design, which can be varied for other types of designs. The examples of other types will be discussed in Section C.



Step 3: By writing the matrix as a system of linear equations, it can readily be seen which symbols should be transmitted during any specific symbol interval.

Step 4: The estimated received symbol vector  $\underline{\tilde{x}}$  is obtained by multiplying Equation (2.12) by  $H^H$  as follows

$$\underline{\tilde{x}} = H^H \underline{y} = H^H H \underline{x} + \underline{\eta}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_t \end{bmatrix} = H^H \begin{bmatrix} y_1 \\ y_2^* \\ y_3 \\ y_4^* \\ y_5 \\ \vdots \\ y_{(t^2-t+2)/2}^* \end{bmatrix} = \begin{bmatrix} h_1^* & -h_2 & -h_3 & -h_4 & \dots & 0 \\ h_2^* & h_1 & 0 & 0 & \dots & 0 \\ h_3^* & 0 & h_1 & 0 & \dots & 0 \\ h_4^* & 0 & 0 & h_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ h_{t-1}^* & 0 & 0 & 0 & \dots & -h_t \\ h_t^* & 0 & 0 & 0 & \dots & h_{t-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2^* \\ y_3 \\ y_4^* \\ y_5 \\ \vdots \\ y_{(t^2-t+2)/2}^* \end{bmatrix} + \underline{\eta} \quad (2.13)$$

where the noise terms are obtained from  $\underline{\eta} = H^H \underline{n}$ .

Step 5: From (2.13), we can show that each estimated symbol is given by

$$\begin{aligned} \tilde{x}_1 &= \left( |h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2 + \dots + |h_t|^2 \right) x_1 + \eta_1 \\ \tilde{x}_2 &= \left( |h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2 + \dots + |h_t|^2 \right) x_2 + \eta_2 \\ &\vdots \\ \tilde{x}_t &= \left( |h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2 + \dots + |h_t|^2 \right) x_t + \eta_t \end{aligned} \quad (2.14)$$

The flow chart in Figure 4 summarizes the design process.

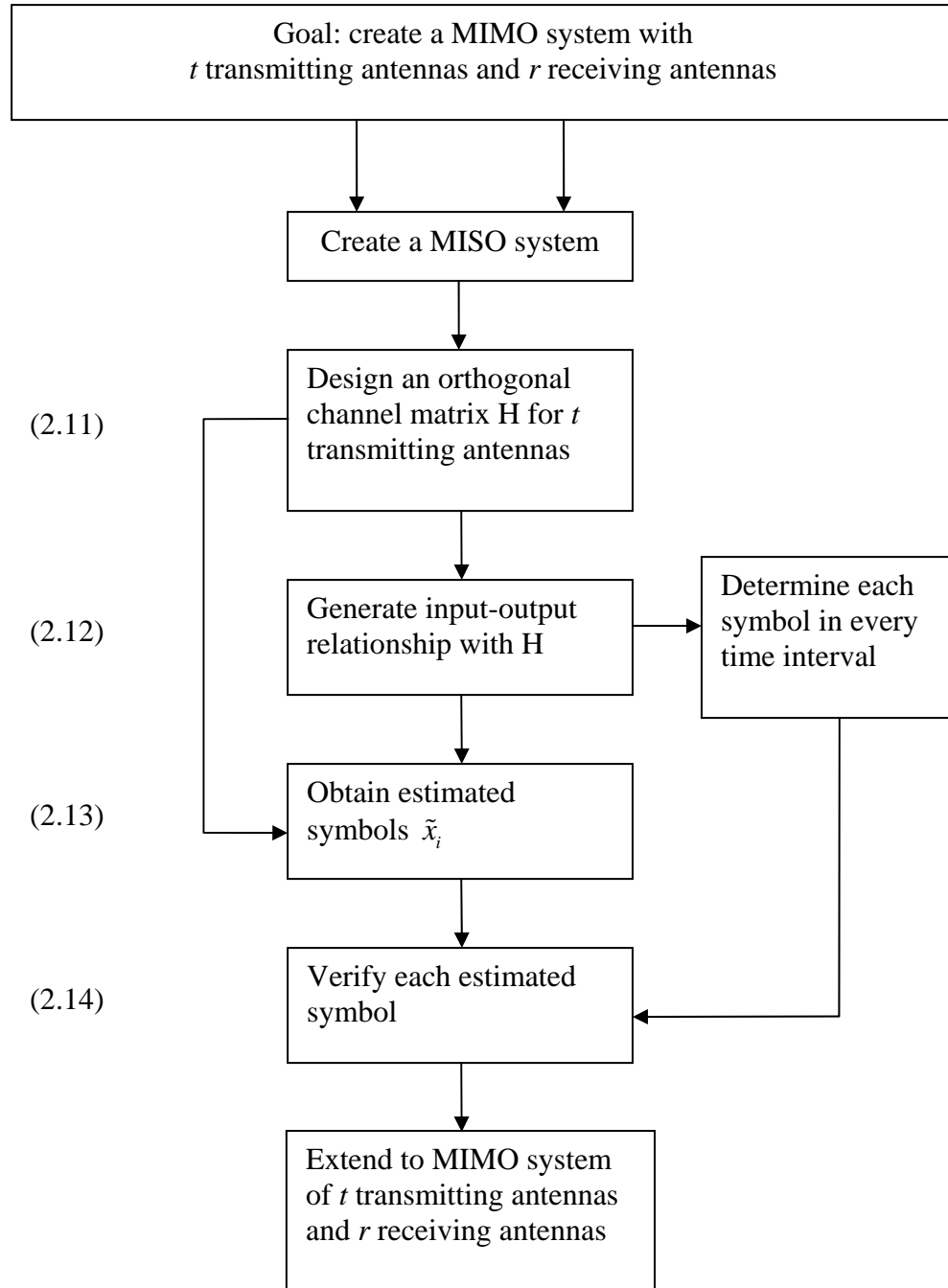


Figure 4. Flowchart of Design Process.

### 3. Generalized Rate Formula

Rate is an important parameter of a MIMO system since it impacts the system's cost. It is related to how much memory the system needs to store the symbols and the overall system complexity. We desire that a system operate at a higher rate and that the magnitude of the recovered signals be as high as possible. As a practical matter, there are tradeoffs between these two desirable properties, and how these competing properties are achieved in different MIMO STBC designs will be examined later.

The rate is defined as the ratio of the number of symbols that can be transmitted to the number of symbol intervals needed. In terms of the channel matrix, the rate is the ratio of the number of columns to the number of rows, since the number of columns represents the number of transmitting antennas which is equal to the number of symbols sent, and the number of rows represents the number of symbol intervals needed.

An easy way to understand the rate is from the orthogonality point of view. In the designing process, from the second row to the end, we inserted certain values into rows in order to make each column pair to be orthogonal. Thus, the total number of rows is the number of column pairs plus one (to account the first row).

The number of column pairs is obtained by choosing a pair from  $t$  as

$$C_2^t = \frac{t(t-1)}{2 \times 1} = \frac{t(t-1)}{2}$$

so the total number of rows is

$$1 + C_2^t = 1 + \frac{t(t-1)}{2 \times 1} = 1 + \frac{t(t-1)}{2} = \frac{t^2 - t + 2}{2}$$

which is shown in Equation (2.12). Thus, the rate of  $t$  antenna system is

$$r_t = \frac{t}{\left(\frac{t^2 - t + 2}{2}\right)} = \frac{2t}{t^2 - t + 2} \quad (2.15)$$

From the rate formula, it can be seen that if  $t > 4$  (i.e., we transmit more than four symbols as a block), the rate drops to less than  $\frac{1}{2}$ , and the rate decreases as the number of symbols transmitted per block increases.

#### 4. Example: Design of a 4×1 System

To illustrate the design process, we build a 4×1 system using the method described above.

In step 1, we begin by writing the first row of the channel matrix with the number of elements equal to the number of transmitting antennas ( $t = 4$ ):

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}$$

We then fill in all the rows to make the columns orthogonal with one another, thereby yielding a 7×4 channel matrix

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_2^* & -h_1^* & 0 & 0 \\ h_3^* & 0 & -h_1^* & 0 \\ h_4^* & 0 & 0 & -h_1^* \\ 0 & h_3^* & -h_2^* & 0 \\ 0 & h_4^* & 0 & -h_2^* \\ 0 & 0 & h_4^* & -h_3^* \end{bmatrix} \quad (2.16)$$

It can be seen that seven symbol periods are needed to transmit four symbols, so the rate is 4/7.

In step 2, with the channel matrix in hand, the relationship between the input and output symbols can be written as

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4^* \\ y_5^* \\ y_6^* \\ y_7^* \end{bmatrix} = H\underline{x} + \underline{n} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_2^* & -h_1^* & 0 & 0 \\ h_3^* & 0 & -h_1^* & 0 \\ h_4^* & 0 & 0 & -h_1^* \\ 0 & h_3^* & -h_2^* & 0 \\ 0 & h_4^* & 0 & -h_2^* \\ 0 & 0 & h_4^* & -h_3^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3^* \\ n_4^* \\ n_5^* \\ n_6^* \\ n_7^* \end{bmatrix} \quad (2.17)$$

In step 3, by writing Equation (2.17) as individual equations, we can determine which symbols should be transmitted by each antenna during any specific symbol interval  $t_i$ :

$$\begin{aligned}
y_1 &= h_1 x_1 + h_2 x_2 + h_3 x_3 + h_4 x_4 + n_1 & \rightarrow & t_1 = [x_1 \quad x_2 \quad x_3 \quad x_4] \\
y_2^* &= h_2^* x_1 - h_1^* x_2 + n_2^* & \rightarrow & t_2 = [-x_2^* \quad x_1^* \quad 0 \quad 0] \\
y_3^* &= h_3^* x_1 - h_1^* x_3 + n_3^* & \rightarrow & t_3 = [-x_3^* \quad 0 \quad x_1^* \quad 0] \\
y_4^* &= h_4^* x_1 - h_1^* x_4 + n_4^* & \rightarrow & t_4 = [-x_4^* \quad 0 \quad 0 \quad x_1^*] \\
y_5^* &= h_3^* x_2 - h_2^* x_3 + n_5^* & \rightarrow & t_5 = [0 \quad -x_3^* \quad x_2^* \quad 0] \\
y_6^* &= h_4^* x_2 - h_2^* x_4 + n_6^* & \rightarrow & t_6 = [0 \quad -x_4^* \quad 0 \quad x_2^*] \\
y_7^* &= h_4^* x_3 - h_3^* x_4 + n_7^* & \rightarrow & t_7 = [0 \quad 0 \quad -x_4^* \quad x_3^*]
\end{aligned} \tag{2.18}$$

In step 4, by multiplying Equation (2.17) by  $H^H$ , the estimated signal is

$$\tilde{\underline{x}} = H^H y = H^H H \underline{x} + \underline{\eta}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} = H^H \begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4^* \\ y_5^* \\ y_6^* \\ y_7^* \end{bmatrix} = \begin{bmatrix} h_1^* & h_2 & h_3 & h_4 & 0 & 0 & 0 \\ h_2^* & -h_1 & 0 & 0 & h_3 & h_4 & 0 \\ h_3^* & 0 & -h_1 & 0 & -h_2 & 0 & h_4 \\ h_4^* & 0 & 0 & -h_1 & 0 & -h_2 & -h_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4^* \\ y_5^* \\ y_6^* \\ y_7^* \end{bmatrix} \tag{2.19}$$

where we can show that

$$H^H H = (|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) I_4$$

and the noise terms are obtained from

$$\underline{\eta} = H^H \underline{n}$$

In step 5, by carrying out multi-vector multiplication in Equation (2.19), the estimated symbols can successfully be verified to be

$$\begin{aligned}
\tilde{x}_1 &= h_1^* y_1 + h_2 y_2^* + h_3 y_3^* + h_4 y_4^* \\
&= h_1^* (h_1 x_1 + h_2 x_2 + h_3 x_3 + h_4 x_4) + h_2 (h_2^* x_1 - h_1^* x_2) + h_3 (h_3^* x_1 - h_1^* x_3) + h_4 (h_4^* x_1 - h_1^* x_4) + \eta_1 \\
&= |h_1|^2 x_1 + h_1^* h_2 x_2 + h_1^* h_3 x_3 + h_1^* h_4 x_4 + |h_2|^2 x_1 - h_1^* h_2 x_2 + |h_3|^2 x_1 - h_1^* h_3 x_3 + |h_4|^2 x_1 - h_1^* h_4 x_4 + \eta_1 \\
&= (|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) x_1 + \eta_1
\end{aligned}$$

$$\tilde{x}_2 = h_2^* y_1 - h_1 y_2^* + h_3 y_5^* + h_4 y_6^* = (|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) x_2 + \eta_2$$

$$\tilde{x}_3 = h_3^* y_1 - h_1 y_3^* - h_2 y_5^* + h_4 y_7^* = (|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) x_3 + \eta_3 \quad (2.20)$$

$$\tilde{x}_4 = h_4^* y_1 - h_1 y_4^* - h_2 y_6^* + h_3 y_7^* = (|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) x_4 + \eta_4$$

where the noise terms are obtained from

$$\underline{\eta} = H^H \underline{n}$$

We conclude that the 4×1 system designed by using the proposed method is indeed an orthogonal design, with rate 4/7, with a channel matrix given by Equation (2.16), and with estimated symbols given by Equation (2.20). This 4×1 system can then be expanded to any 4× $r$  scheme.

### C. OTHER HIGH RATE CODE DESIGN

In the previous section, we presented a MIMO encoding scheme that has the advantage of providing a greater magnitude in the recovered signal as the number of transmitting antennas increases, which works to improve the BER. The scheme has a shortcoming though: the rate decreases as the number of transmitting antennas increases. There are several different MIMO encoding schemes presented in [5] and [9]. These schemes all meet the requirement of orthogonal design but the rate and the magnitude of the recovered symbols differs. In this section, the same criteria that were used in Section B will be used again to examine the schemes from the literature and to differentiate between them.

#### 1. Extension of the Alamouti Scheme

The Alamouti encoding scheme has the highest rate (one) and successfully decreases the BER by increasing the magnitudes of the recovered symbols. An encoding matrix in [5] uses a concatenated form of the Alamouti scheme. The transmitting matrix is

$$X = \begin{bmatrix} x_1 & -x_2^* & 0 & 0 & x_5 & -x_6^* \\ x_2 & x_1^* & x_3 & -x_4^* & 0 & 0 \\ 0 & 0 & x_4 & x_3^* & x_6 & x_5^* \end{bmatrix} \quad (2.21)$$

which means six symbol intervals are needed to transmit six different symbols with three transmitting antennas. The system is illustrated in Figure 5.

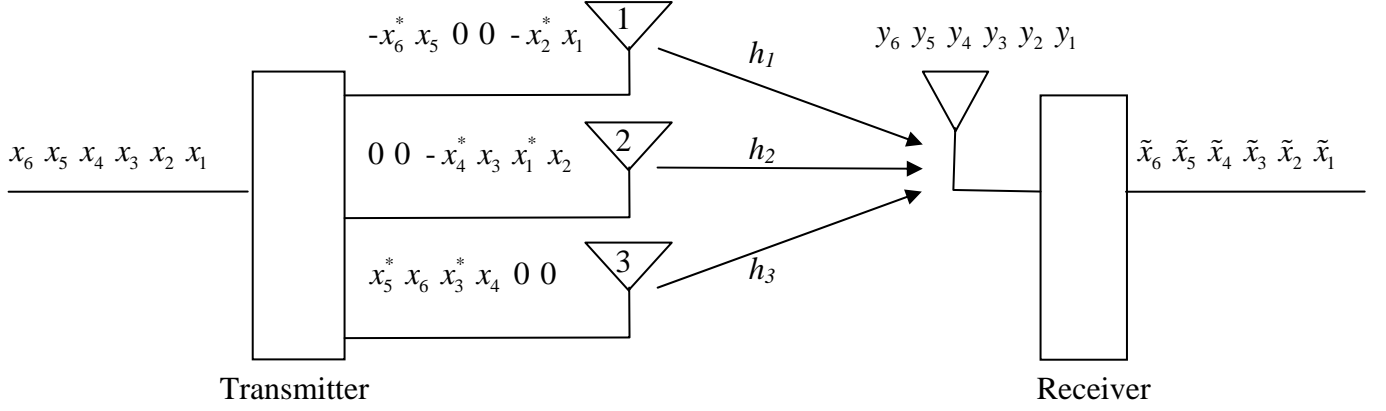


Figure 5. Extension of Alamouti Scheme (After Reference [5].)

From the figure, each reception can be directly written (ignoring the noise terms for convenience) as

$$\begin{aligned}
 y_1 &= h_1 x_1 + h_2 x_2 + n_1 \\
 y_2^* &= -h_1^* x_2 + h_2^* x_1 + n_2^* \\
 y_3 &= h_2 x_3 + h_3 x_4 + n_3 \\
 y_4^* &= -h_2^* x_4 + h_3^* x_3 + n_4^* \\
 y_5 &= h_1 x_5 + h_3 x_6 + n_5 \\
 y_6^* &= -h_1^* x_6 + h_3^* x_5 + n_6^*
 \end{aligned}$$

and the matrix form is

$$\underline{y} = \underline{H}\underline{x} + \underline{n}$$

$$\begin{bmatrix} y_1 \\ y_2^* \\ y_3 \\ y_4^* \\ y_5 \\ y_6^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & 0 & 0 & 0 & 0 \\ h_2^* & -h_1^* & 0 & 0 & 0 & 0 \\ 0 & 0 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & h_3^* & -h_2^* & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_3 \\ 0 & 0 & 0 & 0 & h_3^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \\ n_5 \\ n_6^* \end{bmatrix}$$

Each column of  $H$  is orthogonal to the others; hence this design yields an orthogonal channel matrix with a rate of one.

The estimated symbols can be expressed as

$$\underline{\tilde{x}} = H^H \underline{y} = H^H H \underline{x} + H^H \underline{n}$$

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \\ \tilde{x}_6 \end{bmatrix} = H^H \begin{bmatrix} y_1 \\ y_2^* \\ y_3 \\ y_4^* \\ y_5 \\ y_6^* \end{bmatrix} = \begin{bmatrix} h_1^* & h_2 & 0 & 0 & 0 & 0 \\ h_2^* & -h_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_2^* & h_3 & 0 & 0 \\ 0 & 0 & h_3^* & -h_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1^* & h_3 \\ 0 & 0 & 0 & 0 & h_3^* & -h_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2^* \\ y_3 \\ y_4^* \\ y_5 \\ y_6^* \end{bmatrix}$$

and

$$\begin{aligned} \tilde{x}_1 &= h_1^* y_1 + h_2 y_2^* = h_1^* (h_1 x_1 + h_2 x_2) + h_2 (-h_1^* x_2 + h_2^* x_1) + \eta_1 = (|h_1|^2 + |h_2|^2) x_1 + \eta_1 \\ \tilde{x}_2 &= h_2^* y_1 - h_1 y_2^* = h_2^* (h_1 x_1 + h_2 x_2) - h_1 (-h_1^* x_2 + h_2^* x_1) + \eta_2 = (|h_1|^2 + |h_2|^2) x_2 + \eta_2 \\ \tilde{x}_3 &= h_2^* y_3 + h_3 y_4^* = h_2^* (h_2 x_3 + h_3 x_4) + h_3 (-h_2^* x_4 + h_3^* x_1) + \eta_3 = (|h_2|^2 + |h_3|^2) x_3 + \eta_3 \\ \tilde{x}_4 &= h_3^* y_3 - h_2 y_4^* = h_3^* (h_2 x_3 + h_3 x_4) - h_2 (-h_2^* x_4 + h_3^* x_1) + \eta_4 = (|h_2|^2 + |h_3|^2) x_4 + \eta_4 \\ \tilde{x}_5 &= h_1^* y_5 + h_3 y_6^* = h_1^* (h_1 x_5 + h_3 x_6) + h_3 (-h_1^* x_6 + h_3^* x_5) + \eta_5 = (|h_1|^2 + |h_3|^2) x_5 + \eta_5 \\ \tilde{x}_6 &= h_3^* y_5 - h_1 y_6^* = h_3^* (h_1 x_5 + h_3 x_6) - h_1 (-h_1^* x_6 + h_3^* x_5) + \eta_6 = (|h_1|^2 + |h_3|^2) x_6 + \eta_6 \end{aligned}$$

Comparing these to the result of the Alamouti symbol recovery Equation (2.3), we see that both the traditional Alamouti scheme ( $2 \times 1$  system) and the extension of Alamouti scheme ( $3 \times 1$  system) provide estimates with the same magnitude and rate. Thus, the BER performance should be very similar. The advantage of the extension of Alamouti scheme is that it provides a method for achieving the highest possible rate of one with three antennas. Comparing to the proposed systematic design in Section B, as the number of employed antennas increased, the extension of Alamouti can always achieve the full rate while the rate of the proposed systematic design will decrease. The disadvantages of this extension design are the low magnitude of recovered symbols since most entries of the channel matrix are zero and the corresponding higher BER.



## 2. Rate 1/2 Code with Three Transmitting Antennas

A rate of 1/2 scheme using three transmitting antennas is presented in [9]. The transmitting symbol matrix used is

$$X = \begin{bmatrix} x_1 & -x_2 & -x_3 & -x_4 & x_1^* & -x_2^* & -x_3^* & -x_4^* \\ x_2 & x_1 & x_4 & -x_3 & x_2^* & x_1^* & x_4^* & -x_3^* \\ x_3 & -x_4 & x_1 & x_2 & x_3^* & -x_4^* & x_1^* & x_2^* \end{bmatrix}$$

This scheme needs eight symbol intervals to transmit four different symbols using three transmitting antennas. The rate of 1/2 is smaller than the rate of 3/4 provided by the scheme presented earlier which also used three transmitting antennas (see Section A3). The input-out relation is

$$\underline{y} = H\underline{x} + \underline{n}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5^* \\ y_6^* \\ y_7^* \\ y_8^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & 0 \\ h_2 & -h_1 & 0 & -h_3 \\ h_3 & 0 & -h_1 & h_2 \\ 0 & h_3 & -h_2 & -h_1 \\ h_1^* & h_2^* & h_3^* & 0 \\ h_2^* & -h_1^* & 0 & -h_3^* \\ h_3^* & 0 & -h_1^* & h_2^* \\ 0 & h_3^* & -h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5^* \\ n_6^* \\ n_7^* \\ n_8^* \end{bmatrix}$$

and

$$H^H H = 2(|h_1|^2 + |h_2|^2 + |h_3|^2) I_4$$

and the estimated symbols are

$$\tilde{x}_1 = 2(|h_1|^2 + |h_2|^2 + |h_3|^2)x_1 + \eta_1$$

$$\tilde{x}_2 = 2(|h_1|^2 + |h_2|^2 + |h_3|^2)x_2 + \eta_2$$

$$\tilde{x}_3 = 2(|h_1|^2 + |h_2|^2 + |h_3|^2)x_3 + \eta_3$$

$$\tilde{x}_4 = 2(|h_1|^2 + |h_2|^2 + |h_3|^2)x_4 + \eta_4$$

where the noise terms are obtained from  $\underline{\eta} = H^H \underline{n}$ .

By comparing the rate and the magnitude of the recovered symbols for this design to those of the proposed design in Section B, we conclude that the proposed scheme has the advantage of a lower BER since the magnitude of the recovered signal is double that

of (2.14), but has the disadvantage of a lower rate. We also note that the magnitude of each recovered signal is simply the corresponding diagonal component in the  $H^H H$  matrix. This point can be verified in the  $2 \times 1$ ,  $2 \times 2$ ,  $3 \times 1$ , and  $4 \times 1$  schemes, as well as the extension of the Alamouti scheme and the schemes presented in [6].

### 3. Rate 1/2 Code with Four Transmitting Antennas

Another rate 1/2 design with four transmitting antennas was presented in [9]. The transmitting symbol matrix is

$$X = \begin{bmatrix} x_1 & -x_2 & -x_3 & -x_4 & x_1^* & -x_2^* & -x_3^* & -x_4^* \\ x_2 & x_1 & x_4 & -x_3 & x_2^* & x_1^* & x_4^* & -x_3^* \\ x_3 & -x_4 & x_1 & x_2 & x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & x_3 & -x_2 & x_1 & x_4^* & x_3^* & -x_2^* & x_1^* \end{bmatrix}$$

The input-out relation is

$$\underline{y} = H\underline{x} + \underline{n}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5^* \\ y_6^* \\ y_7^* \\ y_8^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_2 & -h_1 & -h_4 & -h_3 \\ h_3 & -h_4 & -h_1 & h_2 \\ h_4 & h_3 & -h_2 & -h_1 \\ h_1^* & h_2^* & h_3^* & h_4^* \\ h_2^* & -h_1^* & -h_4^* & -h_3^* \\ h_3^* & -h_4^* & -h_1^* & h_2^* \\ h_4^* & h_3^* & -h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5^* \\ n_6^* \\ n_7^* \\ n_8^* \end{bmatrix}$$

and

$$H^H H = 2(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2) I_4$$

and the recovered signals are

$$\begin{aligned}\tilde{x}_1 &= 2\left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right)x_1 + \eta_1 \\ \tilde{x}_2 &= 2\left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right)x_2 + \eta_2 \\ \tilde{x}_3 &= 2\left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right)x_3 + \eta_3 \\ \tilde{x}_4 &= 2\left(|h_1|^2 + |h_2|^2 + |h_3|^2 + |h_4|^2\right)x_4 + \eta_4\end{aligned}$$

This scheme has a fixed rate of 1/2 and an even greater magnitude in the recovered signals, but achieves this improvement at the cost of increased complexity.

This chapter introduced MIMO systems and described their characteristics. A MIMO scheme's transmitting symbol matrix, channel matrix, input-output relationship, rate and recovered symbol were each described in detail, and the need for orthogonal design was emphasized. A systematic way to design a MIMO system was proposed. Three criteria—the rate of the scheme, the BER and the complexity—were used to analyze alternative MIMO designs and compare the advantages and disadvantages of each. It is noted that a system with an advantage under one criterion will tend to have a shortcoming in another aspect. For example, the scheme providing the best BER will sacrifice the rate and the overall complexity.

THIS PAGE INTENTIONALLY LEFT BLANK

### III. CHANNEL MODEL AND ANTENNA SPACING

This chapter briefly reviews the traditional channel model and introduces the parameters that characterize such channels. We then describe the Stanford University Interim (SUI) channel model. In the next chapter, we will use the SUI model to verify the MIMO transmitting schemes presented in the previous chapter. Real-world channels are not memoryless since the same signal may arrive by different paths of varying lengths, and the overall channel impulse response accounts for all of the various paths that a signal may traverse—each with its own magnitude, phase and time delay. Against this backdrop, channel correlation and antenna spacing will be discussed. Under appropriate assumptions and using results from the literature, a plot is presented that displays the relationship between antenna spacing and channel correlation.

#### A. THE FADING CHANNEL

When a signal is transmitted in free space, the direct propagation path from transmitter to receiver is termed the Line-of-Sight (LOS) path. The LOS path is the shortest path, and usually the signal that travels this path contains greater power than the reflected, diffracted or scattered signals that may arrive by other paths. In some environments, the LOS path may not exist, and communication may be sustained only by the signal that arrives via other paths. The arrival of a signal at a receiver via multiple paths is termed the multi-path phenomenon.

The various effects of reflection, scattering, and diffraction of electromagnetic waves in free space may continually change as time progresses, causing unpredictable time-varying changes in the magnitude and phase of the arriving received signal. The phenomenon of multi-path will lead to multiple versions of the same transmitted signal arriving at the receiver at different times. A channel that varies a signal's magnitude and phase as a result of physical effects, such as reflection, diffraction and scattering, is termed a *fading channel* [10].

Two parameters are traditionally used to characterize a fading channel: *coherence time* and *coherence bandwidth*. Coherence time is the statistical measure of the time duration over which a channel is stable or time-invariant. Successful communication

usually requires that the symbol period be less than the coherence time, in which case the channel is said to exhibit *slow fading*. On the other hand, if the duration of the symbol period is greater than the coherence time, the channel is said to exhibit *fast fading* [10].

Coherence bandwidth is the statistical measure of the frequency band over which the channel's frequency response affects all components in the same manner. If the noise-equivalent bandwidth of the signal is less than the coherence bandwidth, then all frequency components will be affected in the same way, and the channel is termed *frequency-nonselective*, or flat. On the other hand, if the noise equivalent bandwidth of the signal that is greater than the coherence bandwidth, the channel is termed *frequency-selective*. One solution for communicating over a frequency-selective channel is to use the Orthogonal Frequency Division Modulation (OFDM) technique [10].

A channel with no LOS path is termed a Rayleigh fading channel. The magnitude of the received signal for such a channel is modeled using a Rayleigh distribution, and the received signal phase is modeled as an uniform random variable in the range between  $-\pi$  and  $\pi$ . A fading channel with a LOS path is termed a Ricean fading channel, and in such cases the received signal magnitude is modeled as a Ricean random variable (the phase is uniformly distributed, as with the Rayleigh case).

A MIMO scheme can actually exploit the characteristics of multi-path to improve performance. By estimating CSI and designing the space-time code properly, the received signals can be reconstructed more accurately. This point will be addressed later in this chapter [6].

## **B. THE SUI CHANNEL**

A channel model useful for broadband fixed wireless applications and based on real measurements was published in 2001. The details of this model, termed the Stanford University Interim (SUI) model, are presented in [8]. The channel model is determined by several factors, such as path loss, delay spread, Doppler spread,  $K$ -factor, and co-channel interference; the  $K$ -factor provides the relative weight between the LOS and fading components. Since these factors, as well as wireless channel behavior, are time-varying, the SUI model used in Matlab simulations only assumes that the mean value is known for each factor.

The SUI channel model describes the path loss due to the terrain type as falling within one of three possible categories: A (moderate-to-heavy), B (intermediate), and C (light tree density). Each of these terrain categories can be further divided into subcategories based on the different combinations of delay spread, Doppler spread, and  $K$ -factor. In [8], the six most common combinations encountered in the wireless environments are chosen as the basis for six different SUI models. Furthermore, each model permits two different choices of an antenna for simulation: an omnidirectional antenna and a  $30^\circ$  directional antenna. The characteristics of the six SUI channels are listed in Table 1 [11].

SUI-Model	Terrain Type	$K$ -factor	Delay Spread	Doppler Spread
SUI-1	C	High	Low	Low
SUI-2	C	High	Low	Low
SUI-3	B	Low	Low	Low
SUI-4	B	Low	Moderate	High
SUI-5	A	Low	High	Low
SUI-6	A	Low	High	High

Table 1. SUI Channel Models and Their Characterization (From Reference [11].)

In Chapter II, the effects of the channel were modeled simply as a complex number representing a magnitude and phase change. With the SUI channel models, the channel, in baseband, considers not only the complex number representing the fading in magnitude and phase, but also considers the effects of delay spread, Doppler spread, and  $K$ -factor. Each channel is represented by a sequence of complex weights; the number of complex weights indicates the length of channel memory. Simulating this channel is much more computationally intensive than was the case for a non-fading channel.

### C. CHANNEL CORRELATION

If the separation between antennas is not sufficiently large with respect to the wavelength of the transmitted signal, the signal cannot be received independently by each antenna; there will be some correlation between channels. In such cases, the advantages of space diversity, such as diversity gain and capacity, will be significantly reduced [1]. The channel correlation can be characterized by a correlation coefficient. In the following section, we will show that the channel correlation is a function of the angles of arrival and departure of the electromagnetic wave, the antenna spacing, the antenna configuration, and the signal wavelength. In this section, useful formulas for modeling a

set of correlated channels will be described, and a method for employing this model with the SUI channel in simulation will be reviewed.

As shown in [11] and [12], the channel matrix describing a set of two correlated fading channels,  $H$ , can be computed from the uncorrelated channel matrix,  $H_w$ , by using the relationship

$$H = R_r^{1/2} H_w R_t^{1/2} \quad (3.1)$$

where  $R_r$  and  $R_t$  are the receiving and transmitting correlation matrices, respectively, and  $R_r^{1/2}$  is the principal square root of  $R_r$ :

$$R_r = R_r^{1/2} R_r^{1/2} \quad (3.2)$$

and similarly

$$R_t = R_t^{1/2} R_t^{1/2}$$

The correlation matrices can be generated as

$$R_r = \begin{bmatrix} 1 & \rho_r \\ \rho_r & 1 \end{bmatrix} \quad (3.3)$$

$$R_t = \begin{bmatrix} 1 & \rho_t \\ \rho_t & 1 \end{bmatrix}$$

where  $\rho_r$  and  $\rho_t$  are the correlation coefficients of the receiving and transmitting antennas, respectively. The values of these coefficients range between zero and one, and depend on several factors, such as the angle of arrival of the electromagnetic wave, the antenna spacing and configuration and the wavelength. This dependence will be discussed in the next section.

If the system order is greater than  $2 \times 2$ , Equation (3.3) can be readily extended to larger orders following the pattern [12]:



$$R_r = \begin{bmatrix} 1 & \rho_r & \rho_r^2 & \dots & \rho_r^n \\ \rho_r & 1 & \rho_r & \dots & \rho_r^{n-1} \\ \rho_r^2 & \rho_r & 1 & \dots & \rho_r^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_r^n & \rho_r^{n-1} & \rho_r^{n-2} & \dots & 1 \end{bmatrix}$$

$$R_t = \begin{bmatrix} 1 & \rho_t & \rho_t^2 & \dots & \rho_t^n \\ \rho_t & 1 & \rho_t & \dots & \rho_t^{n-1} \\ \rho_t^2 & \rho_t & 1 & \dots & \rho_t^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_t^n & \rho_t^{n-1} & \rho_t^{n-2} & \dots & 1 \end{bmatrix}$$

The channel correlation between antennas that are close together will be greater than that of antennas that are further apart; as the separation between antennas increases, the correlation will decrease.

In the case of a Ricean fading channel, the actual MIMO channel matrix is adjusted by the addition of a constant term that accounts for the mean value of the effects of the LOS component:

$$H = \sqrt{\frac{K}{K+1}} \bar{H} + \sqrt{\frac{1}{K+1}} H_w \quad (3.4)$$

where  $\bar{H}$  is derived by considering the LOS transmissions with unit power, and  $H_w$  considers the effect of the fading channels. The Ricean  $K$ -factor is the variable that adjusts the relative weight of the terms that account for the LOS and fading components. If the channel is a Rayleigh faded channel,  $K$  will be zero; if there is no fading,  $K$  will be infinity; and if there is Ricean fading, the  $K$ -factor will be between zero and infinity. Figure 6 show a conceptual diagram [11] outlining the procedure for generating a correlated SUI channel model.

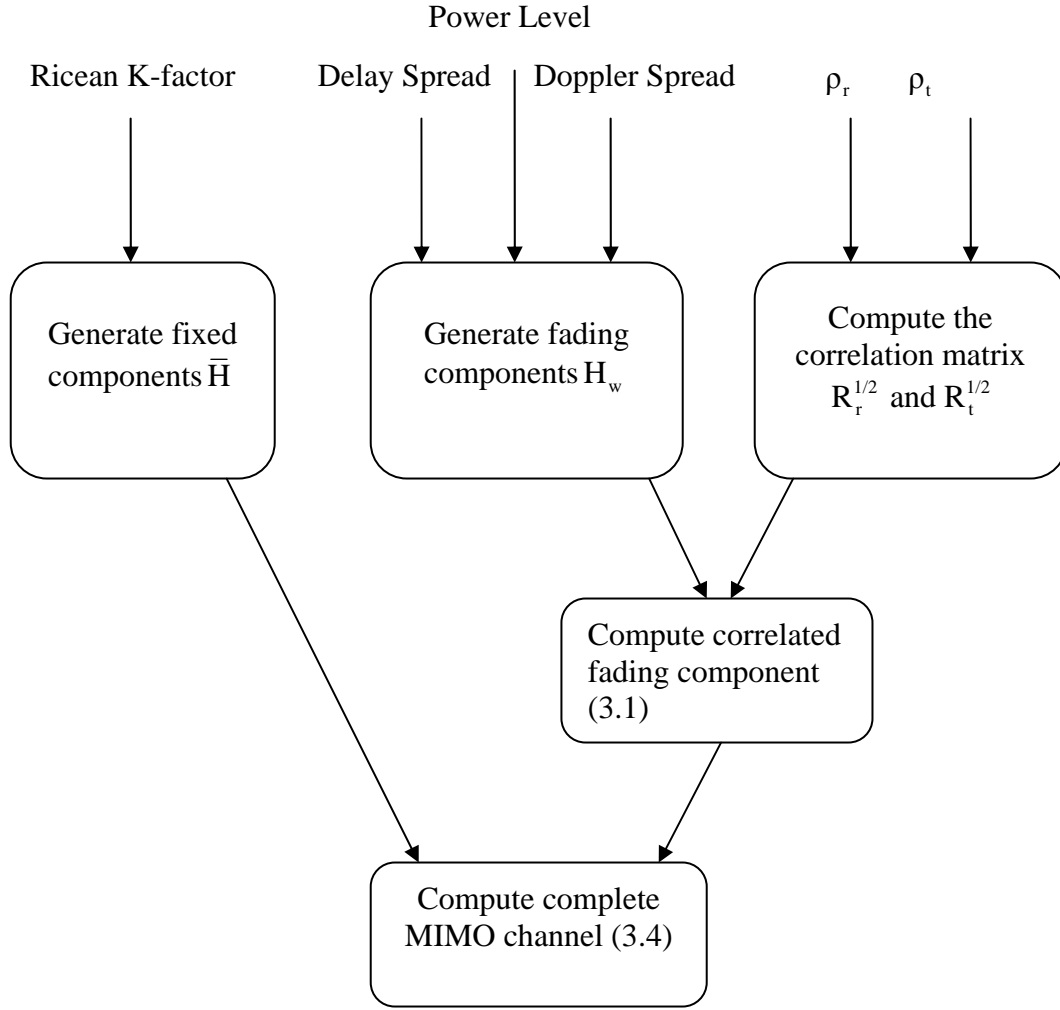


Figure 6. Conceptual Diagram for Generating a Correlated MIMO Channel (After Reference [11].)

#### D. ANTENNA SPACING AND CORRELATION COEFFICIENT

As mentioned previously, the correlation coefficients depend on the angle of arrival of the electromagnetic waves, the antenna spacing, the antenna configuration, and the signal wavelength. Closed-form solutions for calculating the correlation coefficients are presented in [1] and [2]. These results will be used in the present discussion.

As shown in [1], if all the antennas elements and incoming signal waves are in the x-y plane, the correlation coefficient can be determined using

$$\rho_{m,n} = \frac{E\{r_m(t)r_n^*(t)\}}{E\{r_n(t)r_m^*(t)\}} \quad (3.5)$$

where  $E\{\bullet\}$  is the expectation operator and  $r_m(t)$  is the signal received by the  $m^{th}$  antenna.

Equation (3.5) can be calculated as

$$\rho_{m,n} = \int_{\alpha} e^{-j2\pi \frac{D(m,n)}{\lambda} \cos(\alpha - \theta_{m,n})} f(\alpha) d\alpha \quad (3.6)$$

where  $\lambda$  is the wavelength,  $D(m,n)$  is the distance function between  $m^{th}$  and  $n^{th}$  antennas,  $\theta_{m,n}$  is the angle between the x axis and the vector connecting  $m^{th}$  to  $n^{th}$  antennas,  $\alpha$  is the angle of arrival of the incoming wave, and  $f(\alpha)$  is the probability density function of angle of arrival (AOA) of the incoming wave. Figure 7 illustrates the antenna geometry.

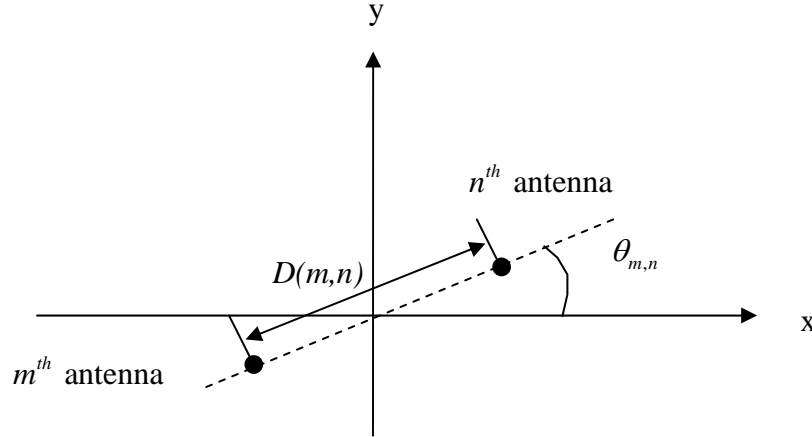


Figure 7. Antenna Configuration.

Under the assumption that the scattering signals are independent of each other, a general closed-form expression for the real and imaginary parts of the fading correlation function is given by [1] as

$$\text{Re}\{\rho_{m,n}\} = J_0(2\pi \frac{D(m,n)}{\lambda}) + 2 \sum_{k=1}^{\infty} (-1)^k J_{2k}(2\pi \frac{D(m,n)}{\lambda}) \times \int_{\theta} \cos[2k(\alpha - \theta_{m,n})] f(\alpha) d\alpha \quad (3.7)$$

$$\text{Im}\{\rho_{m,n}\} = 2 \sum_{k=1}^{\infty} (-1)^k J_{2k+1}(2\pi \frac{D(m,n)}{\lambda}) \times \int_{\theta} \cos[(2k+1)(\alpha - \theta_{m,n})] f(\alpha) d\alpha \quad (3.8)$$

where  $J_n(x)$  is the  $n^{\text{th}}$  order Bessel function of first kind and  $n$  is a integer.

In a uniform linear antenna array, where all antenna elements are arranged in a line on the x axis with a constant spacing of  $d$  m, the distance function can be simplified (into just the distance along the x-axis between antennas) and angle  $\theta_{m,n}$  is simply zero as illustrated in Figure 7. In this scenario, Equation (3.6) can be simplified as

$$\rho_{m,n} = \int_{\alpha} e^{-j2\pi \frac{d}{\lambda} \cos(\alpha)} f(\alpha) d\alpha \quad (3.9)$$

In this case, the correlation coefficient is simply a function of the constant spacing between adjacent antenna elements, the array configuration, the wavelength, and the distribution of the signal's AOA. The simplified configuration of the uniform linear array is shown in Figure 8, with antenna element spacing equal to  $d$  and angle  $\theta_{m,n} = 0$ .

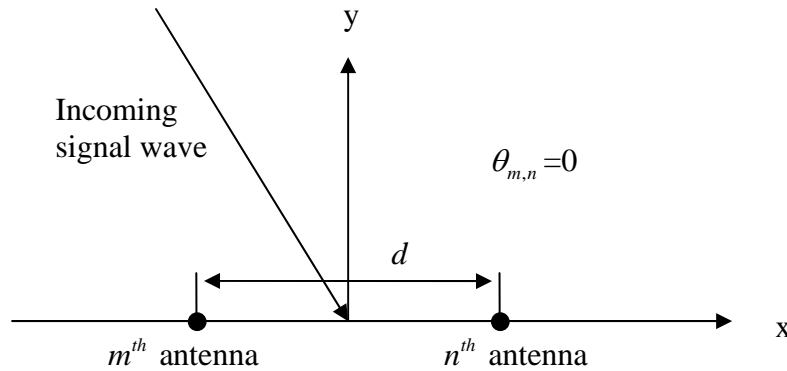


Figure 8. Simplified Antenna Configuration of a Uniform Linear Array.

If the antenna configuration is not the special case of a linear array, then the distance function  $D(m,n)$  and the angle  $\theta_{m,n}$  will need to be recomputed.

It has been noted that the correlation coefficients depend on the probability distribution of the AOA of the incoming wave. We next discuss in detail two different distribution functions that are useful to characterize the AOA. Following this, the relationship between the correlation coefficients and the antenna element spacing will be examined in greater depth, and simulation results will be presented.

### 1. Laplacian Distribution

In [2], the general MIMO system configuration of Figure 9 is analyzed under the assumptions of small scale area (i.e., the signal suffers no path loss between antenna elements), omni-directional antenna employment and far-field conditions. The far-field condition implies that the distance between the transmitter and receiver is far apart so that the arriving wave can be approximated as a plane wave [13]. For the configuration shown in Figure 9, [2] presents a closed-form solution for the correlation coefficients.

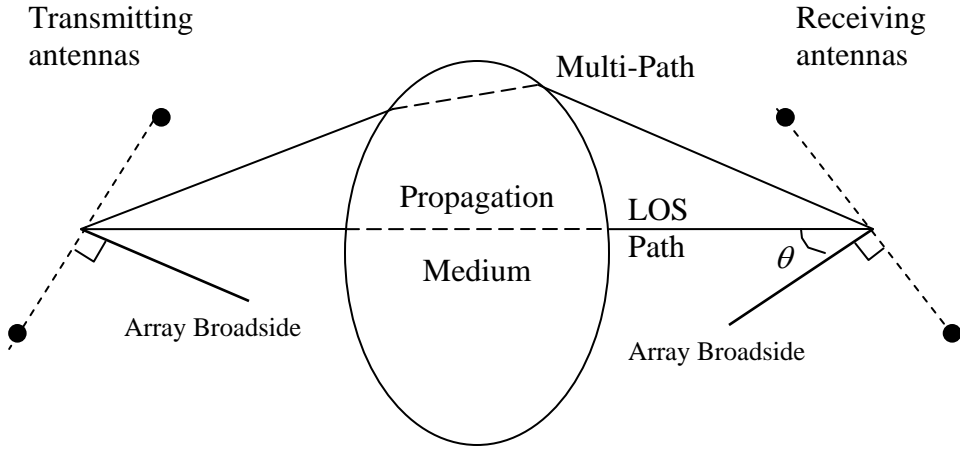


Figure 9. MIMO System Configuration.

Using a Laplacian distribution to characterize the AOA is one of the widely-used models for MIMO systems [2]. The Laplacian probability density function is of the form

$$f_A(\alpha) = \frac{1}{\sqrt{2}\sigma_A} e^{-\frac{\sqrt{2}}{\sigma_A}|\alpha|}, \quad \alpha \in [-\pi, \pi] \quad (3.10)$$

where  $\sigma_A$  is the standard deviation and the  $\theta$  is the angle between the array broadside and the LOS path. The geometry is presented pictorially in Figure 10.

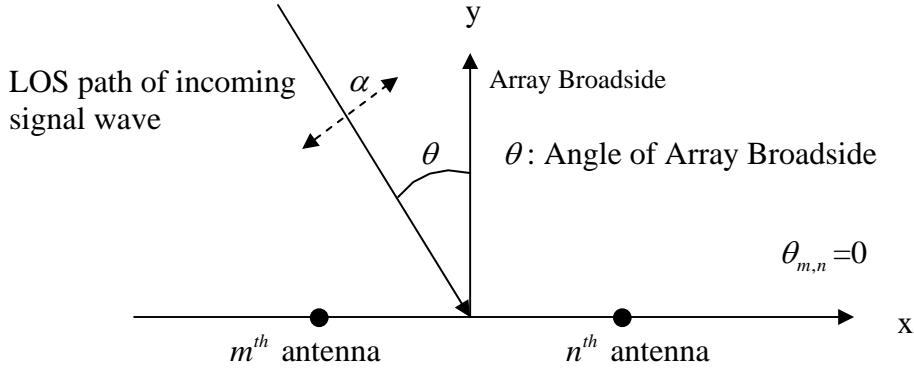


Figure 10. Conceptual Diagram of Angle of Arrival.

Substituting Equation (3.10) into Equation (3.6), the correlation values are computed as

$$\begin{aligned}
 \rho(d) &= \int_{\alpha} e^{-j2\pi \frac{d}{\lambda} \cos(\alpha - \theta_{m,n})} f(\alpha) d\alpha \\
 &= \frac{1}{\sqrt{2}\sigma_A} \int_{-\pi}^{\pi} e^{-j2\pi \frac{d}{\lambda} \cos(\alpha - \theta_{m,n})} e^{\frac{\sqrt{2}}{\sigma_A} |\alpha|} d\alpha \\
 &= \frac{1}{\sqrt{2}\sigma_A} \int_{-\pi}^{\pi} e^{\frac{\sqrt{2}}{\sigma_A} |\alpha|} e^{-jkd \cos(\alpha)} d\alpha
 \end{aligned} \tag{3.11}$$

where  $k = \frac{2\pi}{\lambda}$  and  $\theta_{m,n} = 0$ . Equation (3.11) can be rewritten using  $n^{th}$  order Bessel functions of the first-kind as [2]

$$\begin{aligned}
 \rho(d) &\approx \left[ J_0(kd) + \sum_{n=1}^{\infty} \frac{4 \cos(2n\theta)}{\sigma_A^2 (2n)^2 + 2} J_{2n}(kd) \right] \\
 &\quad + j \left[ \sum_{n=1}^{\infty} \frac{4 \sin[(2n-1)\theta]}{\sigma_A^2 (2n-1)^2 + 2} J_{2n-1}(kd) \right]
 \end{aligned} \tag{3.12}$$

The approximation of magnitudes of correlation coefficient in (3.12) are computed and shown in Figures 11 and 12 with the summation index  $n$  from 1 to 40 for two  $\sigma_A$  values.

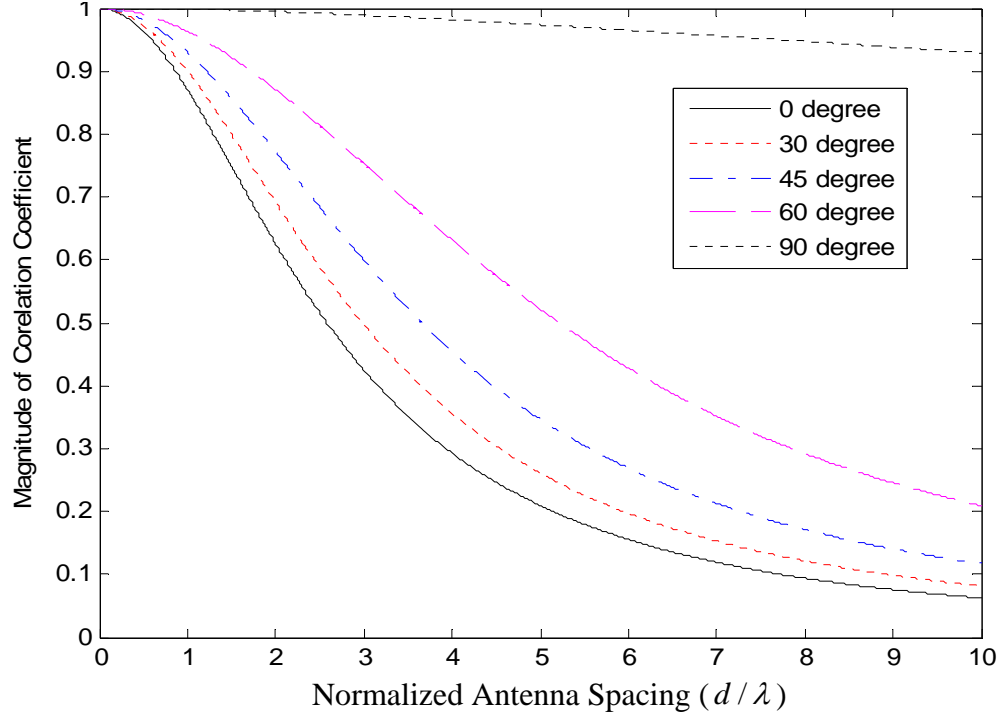


Figure 11. Magnitude of Correlation Coefficient versus Normalized Antenna Spacing for Different  $\theta$  (Laplacian with  $\sigma_A = 5^\circ$ , From Reference [2].)

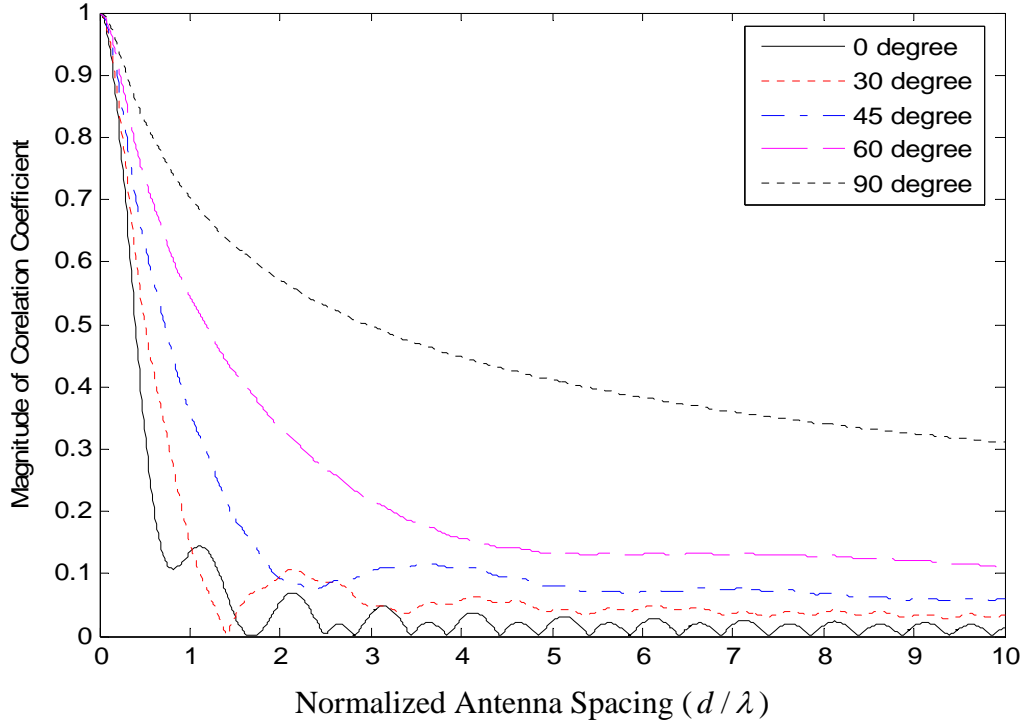


Figure 12. Magnitude of Correlation Coefficient versus Normalized Antenna Spacing for Different  $\theta$  (Laplacian with  $\sigma_A = 35^\circ$ , From Reference [2].)

Several conclusions can be drawn by examining Figures 11 and 12. First, it is evident that  $\theta = 0^\circ$  is the best case scenario, and as the value of  $\theta$  increases, the correlation coefficient increases. This observation makes sense since  $\theta$  is the angle between the LOS path of incoming signal wave and the array broadside. Intuitively, it is clear that  $\theta = 90^\circ$  should comprise the worst case.

Second, by comparing the  $\theta = 0^\circ$  curves in Figures 11 and 12, an interesting result is noted. Figure 11 shows that the correlation coefficient first drops to approximately 0.1 at about  $d = 7\lambda$ , while Figure 12 shows that the correlation coefficient first drops to a value of 0.1 when  $d = 0.8\lambda$ . The difference in these two values—almost an order of magnitude—might be quite considerable if the wavelength is not small. This disparity can be explained by noting that the larger standard deviation  $\sigma_A = 35^\circ$  used in Figure 12 represents a larger number of multi-path arrivals, and the small standard deviation  $\sigma_A = 5^\circ$  used in Figure 11 implies only a few multipath arrivals. The bottom line is that the greater the number of multi-path arrivals, the less correlated the channels will be. A MIMO system can take advantage of the large number of multi-path arrivals to achieve better performance.

Third, if a MIMO system exhibits greater multi-path, it will be less sensitive to the effects of  $\theta$ . This is readily observed in Figures 11 and 12. Examining the worst case situation when  $\theta$  is 90 degrees, we note that the performance in Figure 12 is considerably better than that of Figure 11, because the system of Figure 12 has more possible communication paths between sender and receiver, and hence more paths that can potentially serve to maintain communication.

## 2. Uniform Distribution

In the case that the AOA is modeled using a uniformly distributed probability density function, Equations (3.7) and (3.8) can be expressed as [2]



$$\begin{aligned}
\text{Re}\{\rho_{m,n}\} &= J_0\left(2\pi \frac{D(m,n)}{\lambda}\right) + 2 \sum_{k=1}^{\infty} (-1)^k J_{2k}\left(2\pi \frac{D(m,n)}{\lambda}\right) \times \\
&\quad \int_{\theta} \cos[2k(\alpha - \theta_{m,n})] f(\alpha) d\alpha \\
&= J_0(kd) + 2 \sum_{k=1}^{\infty} (-1)^k J_{2k}(kd) \times \int_{-\pi}^{\pi} \frac{1}{2\pi} \cos(2k\alpha) d\alpha \\
&= J_0(kd)
\end{aligned}$$

and

$$\begin{aligned}
\text{Im}\{\rho_{m,n}\} &= 2 \sum_{k=1}^{\infty} (-1)^k J_{2k+1}\left(2\pi \frac{D(m,n)}{\lambda}\right) \times \\
&\quad \int_{\theta} \cos[(2k+1)(\alpha - \theta_{m,n})] f(\alpha) d\alpha \\
&= 2 \sum_{k=1}^{\infty} (-1)^k J_{2k+1}(kd) \times \int_{-\pi}^{\pi} \frac{1}{2\pi} \cos[(2k+1)\alpha] d\alpha \\
&= 0
\end{aligned}$$

The correlation coefficient for the case of the uniform distribution is real and exhibits periodic zeroes due to the Bessel function. The correlation coefficient versus normalized wavelength is shown in Figure 13. By comparing Figure 13 with Figure 11 and 12, it is evident that the distribution of the AOA greatly affects the correlation coefficient.

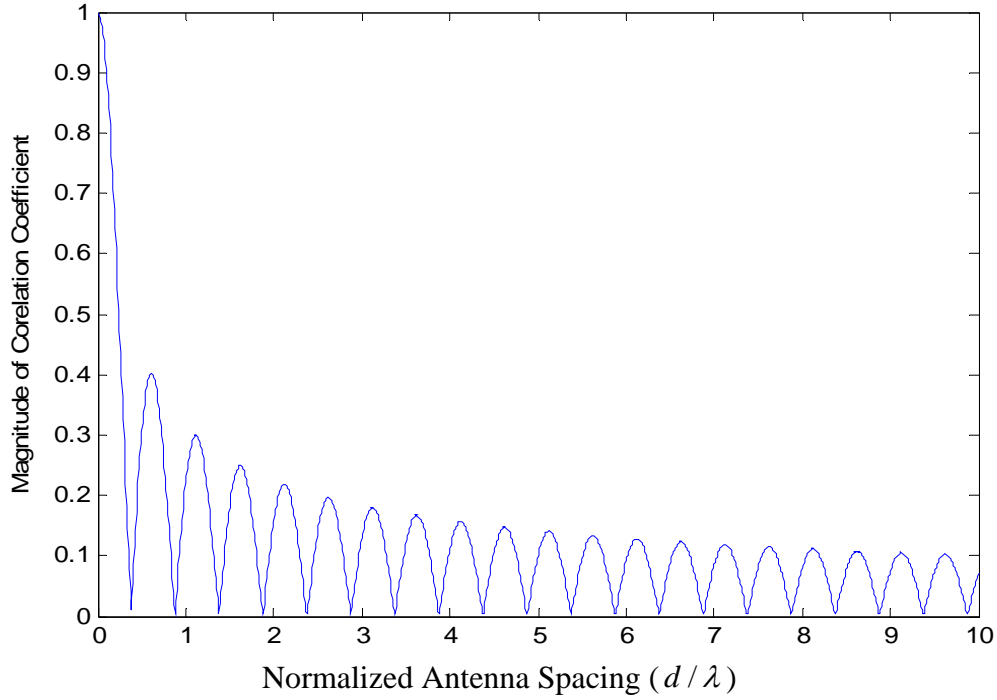


Figure 13. Magnitude of Correlation Coefficient versus Normalized Antenna Spacing (Uniform Distribution from  $-\pi$  to  $\pi$ , From Reference [2].)

From the discussion above (and from points further elaborated on in [14]), it can be readily seen that the accurate measurement of the signal's AOA distribution is crucial to the design of a MIMO system. Many different AOA distributions are presented in the literature; for example, an AOA modeled as a Gaussian distribution is discussed in [7].

### 3. Antenna Spacing

Let us consider a widely used wireless networking standard, the IEEE 802.11, operating in the 2.4-GHz band to determine the correlation coefficients. The wavelength in this case can be calculated as

$$\lambda = C / f = 3 \times 10^8 / 2.4 \times 10^9 = 0.125 \text{ m}$$

Given the wavelength, the magnitude of the correlation coefficient is plotted as a function of antenna spacing for different values of  $\theta$  and  $\sigma_A$  in Figures 14 and 15 for Laplacian AOA distribution.

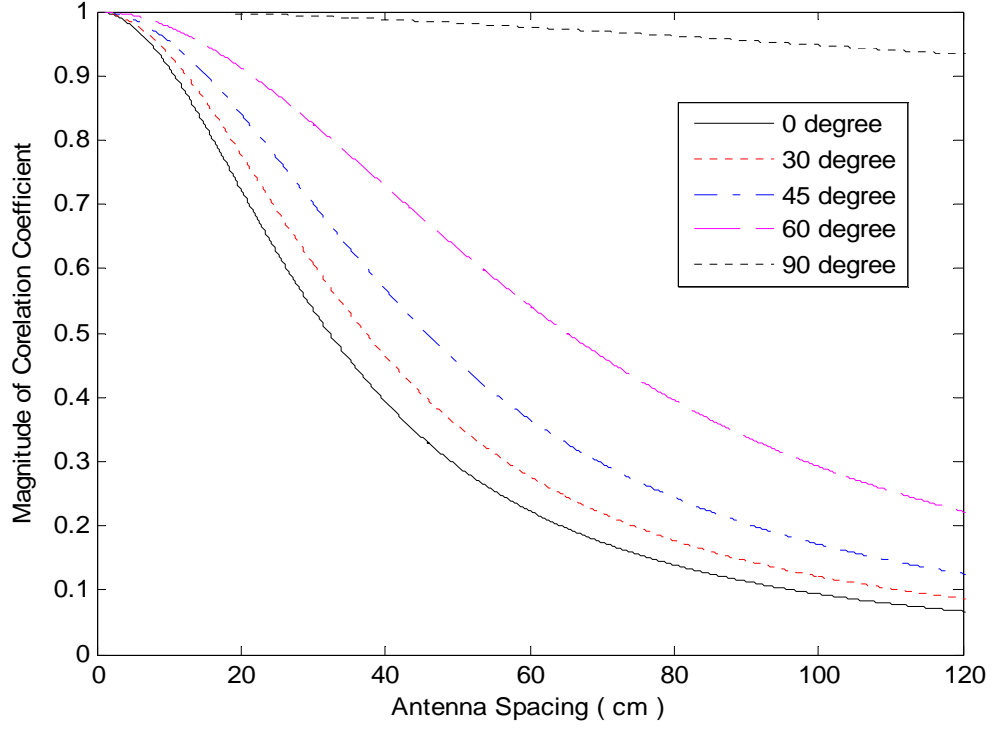


Figure 14. Magnitude of Correlation Coefficient versus Antenna Spacing for Different  $\theta$  (Laplacian with  $\sigma_A = 5^\circ$ , After Reference [2].)

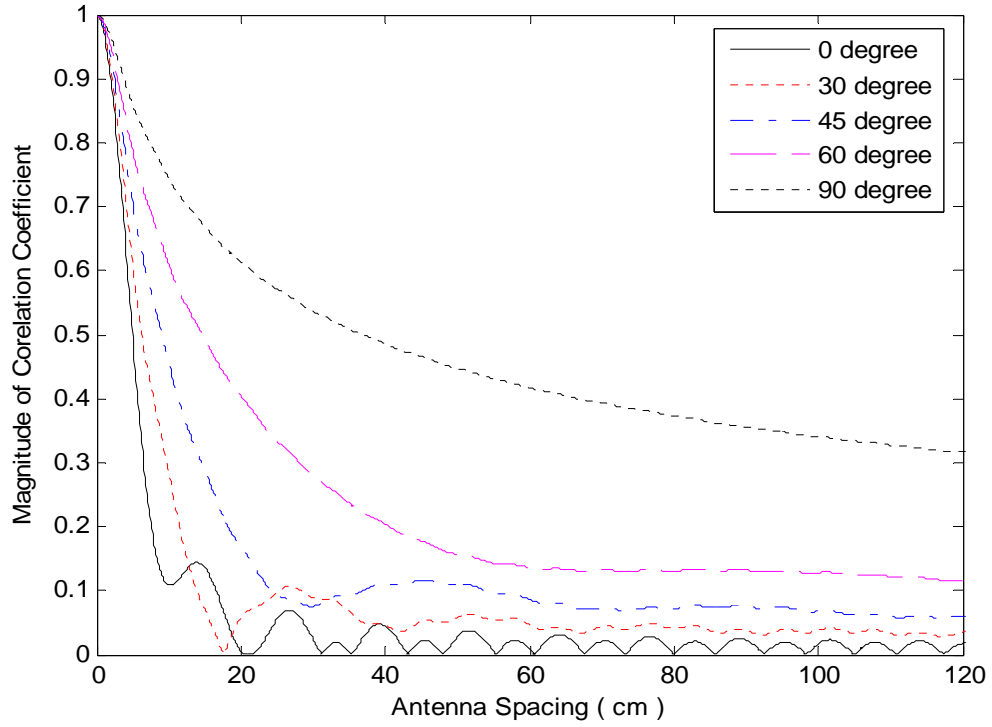


Figure 15. Magnitude of Correlation Coefficient versus Antenna Spacing for Different  $\theta$  (Laplacian with  $\sigma_A = 35^\circ$ , After Reference [2].)

The correlation coefficient of a MIMO system can be physically related to the antenna spacing. Figures 14 and 15 show how the correlation coefficient in an IEEE802.11-based system varies with actual antenna spacing for a given transmitting frequency. The greater spacing tends to reduce the correlation between antennas and can obviously improve the performance. On the other hand, greater spacing is not desirable since the physical size of the receiver will be even bigger.

The performance of a MIMO system is affected by antenna correlation. If a MIMO system is required to perform with a certain BER, many different designs employing different combinations of transmitting and receiving antennas can be considered. When accounting for the degradation of performance due to correlated antenna elements, the actual spacing needs to be chosen appropriately so that desirable performance will be maintained. This is the key connection between Chapter II and Chapter III. This point will be further developed through simulation studies in the next chapter.

In this chapter, the basic concepts of fading, multi-path, the MIMO channel, and antenna spacing were reviewed. Using the results from the published literature, a foundation for simulating a MIMO channel was developed. The concept of channel correlation was introduced and applied to the MIMO channel. Various models that describe the angle of arrival of the received signal were discussed and applied to the calculation of channel correlation. The results presented in this chapter will be verified through simulation in the next chapter, and an interesting connection between performance and necessary antenna spacing will be presented. A MIMO system designer must be equipped with these fundamentals in order to ensure that the actual results will meet the desired performance requirements.

## IV. SIMULATION RESULTS

This chapter will use the theory developed in Chapters II and III, as well as the systematic MIMO design approach proposed in Chapter II, to study the BER performance of various MISO and MIMO schemes in a Rayleigh fading channel and in a SUI channel. All the simulations are performed using MATLAB. Simulation studies for MIMO systems of orders  $2 \times 1$  and  $2 \times 2$  were presented in [11]. By using the theory presented in Chapters II and III, we explain how systems of higher order can be evaluated in simulation and present simulation results for systems up to an order of  $4 \times 4$ . We then present simulation results that show the effects of channel correlation on the BER performance of a  $4 \times 1$  MISO scheme in a SUI channel and study the BER performance of MISO transmitters that consist of uniform linear antenna arrays with random inter-antenna spacing. The results of an application of the latter to sensor networks are presented.

### A. BASEBAND MODEL

The analysis of bandpass communication systems is simplified by considering the equivalent baseband model, which is the lowpass equivalent of the bandpass system. The baseband model contains all the necessary communication signal information, but neglects terms that relate strictly to the carrier frequency, yielding mathematical expressions that are easier to work with. The real input bandpass signal  $N(t)$  can be represented in terms of a complex baseband signal  $Z(t)$  as [15]

$$\begin{aligned} N(t) &= \text{Re}[Z(t)\exp(j\omega_c t)] \\ &= A(t)\cos[\omega_c t + \theta_0(t)] \\ &= X(t)\cos\omega_c t - Y(t)\sin\omega_c t \end{aligned} \tag{4.1}$$

where  $X(t) = \text{Re}[Z(t)]$ ,  $Y(t) = \text{Im}[Z(t)]$ ,  $A(t) = |Z(t)|$ , and  $\theta_0(t) = \angle Z(t)$ . The equation shows that the bandpass signal  $N(t)$  can be represented in three different ways.

In Quadrature Phase Shift Keying (QPSK), the transmitted signal is given by

$$s(t) = \sqrt{2}A_c \cos(2\pi f_c t + \theta_m)$$

where the signal can take on one of four possible values, corresponding to four possible choices of the phase angle:  $\theta_m = \pi/4, 3\pi/4, -\pi/4, -3\pi/4$ ;  $A_c$  denotes the signal amplitude and  $f_c$  is the carrier frequency. This QPSK signal can be rewritten as [10]

$$\begin{aligned} s(t) &= \sqrt{2}A_c \operatorname{Re}\{\exp[j(\omega_c t + \theta_m)]\} \\ &= \sqrt{2}A_c \operatorname{Re}\{\exp(j\theta_m) \exp(j\omega_c t)\} \\ &= \sqrt{2}A_c [\cos(\theta_m) \cos(\omega_c t) - \sin(\theta_m) \sin(\omega_c t)] \end{aligned} \quad (4.2)$$

where  $\omega_c = 2\pi f_c$ . Comparing the last line of Equation (4.1) with the middle line of (4.2), it is noted that  $\sqrt{2}A_c \exp(j\theta_m)$  is the baseband signal corresponding to the (bandpass) QPSK signal,  $s(t)$ . The real and imaginary components of the baseband signal are evident by comparing the last line of (4.1) with the last line of (4.2). Equation (4.2) can be further represented as

$$\begin{aligned} s(t) &= \sqrt{2}A_c [\cos(\theta_m) \cos(\omega_c t) - \sin(\theta_m) \sin(\omega_c t)] \\ &= A_c [d_i(t) \cos(\omega_c t) - d_q(t) \sin(\omega_c t)] \\ &= \operatorname{Re}\{A_c [d_i(t) + jd_q(t)] \exp(j\omega_c t)\} \end{aligned} \quad (4.3)$$

where  $d_i(t)$  is termed the in-phase component of the transmitted signal and  $d_q(t)$  is termed the quadrature component. These two components,  $d_i(t)$  and  $d_q(t)$  take on values of  $\pm 1$ . The four possible values of the baseband signal (4.3) (corresponding to the four possible combinations of  $d_i(t)$  and  $d_q(t)$ ) are represented in a simplified manner using a constellation plot, as shown in Figure 16.

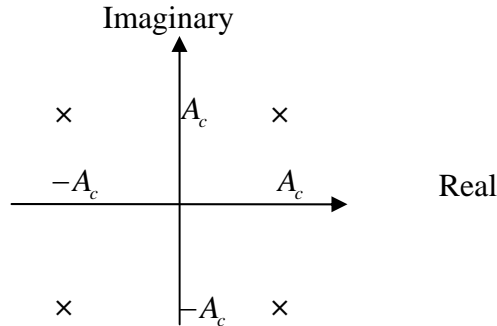


Figure 16. QPSK Constellation Plot.

In this chapter, bandpass signals will be represented by their corresponding baseband signals, and all simulations will be performed using the baseband representations.

## B. PERFORMANCE OF A MISO SYSTEM OVER A RAYLEIGH FADING CHANNEL

The first simulation evaluates the performance of a MISO system over a Rayleigh fading channel. The systematic design methodology presented in Chapter II is used to expand an Alamouti scheme to systems that employ a higher number of transmitting antennas. Specifically, in this section, the BER performance of  $1 \times 1$ ,  $2 \times 1$ ,  $3 \times 1$ , and  $4 \times 1$  systems will be compared as a function of energy per bit to noise power spectral density ratio,  $E_b / N_o$ . Recall that as the number of antennas increases, the transmitted data will be encoded into different blocks of signals, using STBC as described in Chapter II. Consequently, different reception and recovery computations are applied to decode each block. The complexity of the required computations obviously increases as the number of transmitting antennas increases.

In this simulation, the channel is modeled as a slowly-fading Rayleigh channel in which the signal period is less than the coherence time, as described in Chapter III. The receiver has perfect CSI to generate estimates for the recovered signal, as described in Chapter II. No additional channel coding is used, and all channels are uncorrelated. A single carrier frequency scheme employing QPSK modulation, as described in the previous section, is used for transmission. Each antenna transmits the same amount of power. The bit streams are generated randomly and mapped to QPSK symbols, which are then rearranged into STBC blocks for transmission over the Rayleigh fading channel. The receiver decodes received blocks back to information symbols. After the information symbols are mapped back to a bit stream, the BER is computed. The details of the simulation are listed in Table 2.

# of transmitted information bits for each $E_b / N_o$	24,000,000
# of symbols transmitted for each $E_b / N_o$	12,000,000
# of Monte Carlo runs	12

Table 2. Simulation of a MISO System over a Rayleigh Fading Channel.

Figure 17 shows the performance of the four MISO systems over a Rayleigh fading channel. The Rayleigh random variable is generated by taking the square root of

the sum of two independent zero-mean and unit-variance Gaussian random variables, and applying the result to the real and imaginary parts of the baseband signal. The  $1 \times 1$  and  $2 \times 1$  curves match exactly with simulation results presented in [11]. The BER improves significantly (over 10 dB) as the system is changed from  $1 \times 1$  to  $2 \times 1$ . As the number of transmitting antennas increases, the performance continues to improve, but each time to a lesser degree. These results conform to the analysis presented in Chapter II, which noted that the magnitude of the recovered signal increases linearly with the number of transmitting antennas (see Equation 2.22), but the BER is not a linear function of the magnitude of the recovered signal [6].

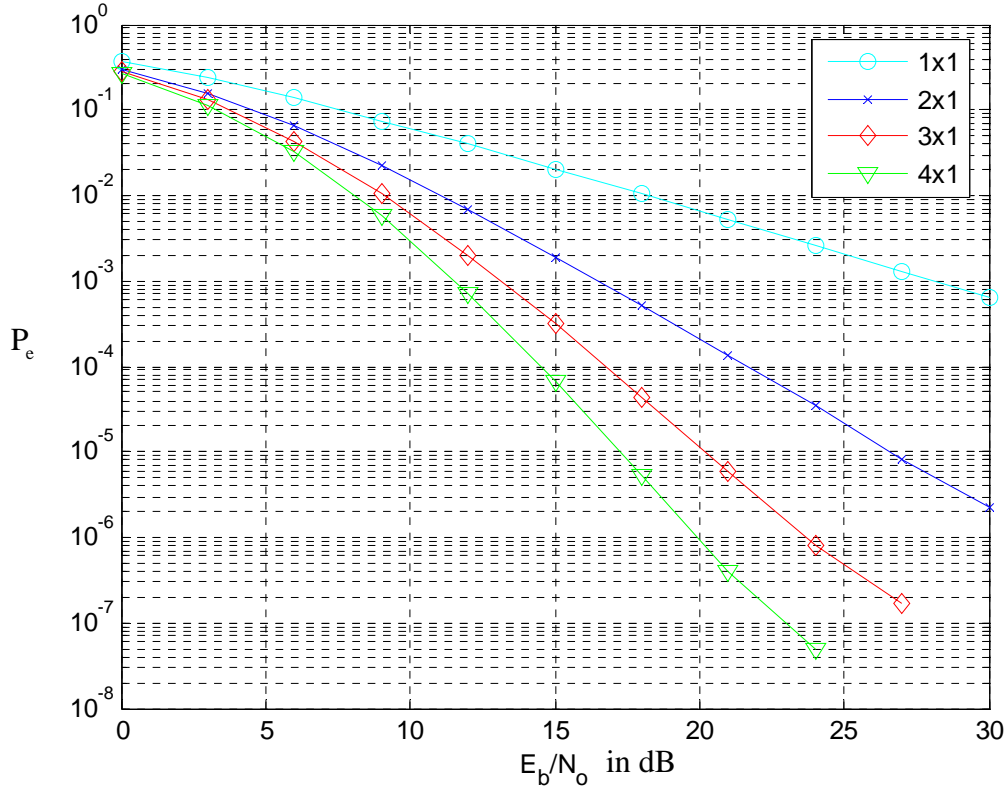


Figure 17. BER Performance of MISO Systems over a Rayleigh Fading Channel.

### C. PERFORMANCE OF A MIMO SYSTEM OVER A RAYLEIGH FADING CHANNEL

In the previous section, we evaluated the BER performance of different systems, each containing a single receive antenna. In this section, we investigate the BER performance of MIMO systems as the number of receiving antennas increases. As in the



previous section, the modulation scheme is QPSK, no channel coding is used, the channel is modeled as a Rayleigh fading channel (in the same manner as the previous section), and all channels are uncorrelated with each other. All the assumptions of the previous simulation are retained, except for the number of receiving antennas, which is allowed to increase. Table 2 summarizes how the simulations were conducted for each evaluated system (i.e.,  $2 \times 1$ ,  $2 \times 2$ ,  $3 \times 1$ , etc.).

Figures 18, 19 and 20 display the BER performance of MIMO systems for two, three and four transmitting antennas, respectively. In each figure, curves are shown indicating the performance for different numbers of receiver antennas (specifically, the number of receiver antennas is varied from one to the number of transmitting antennas). The curves displayed in Figure 18 match similar results presented in [11] for the  $2 \times 1$  and  $2 \times 2$  systems.

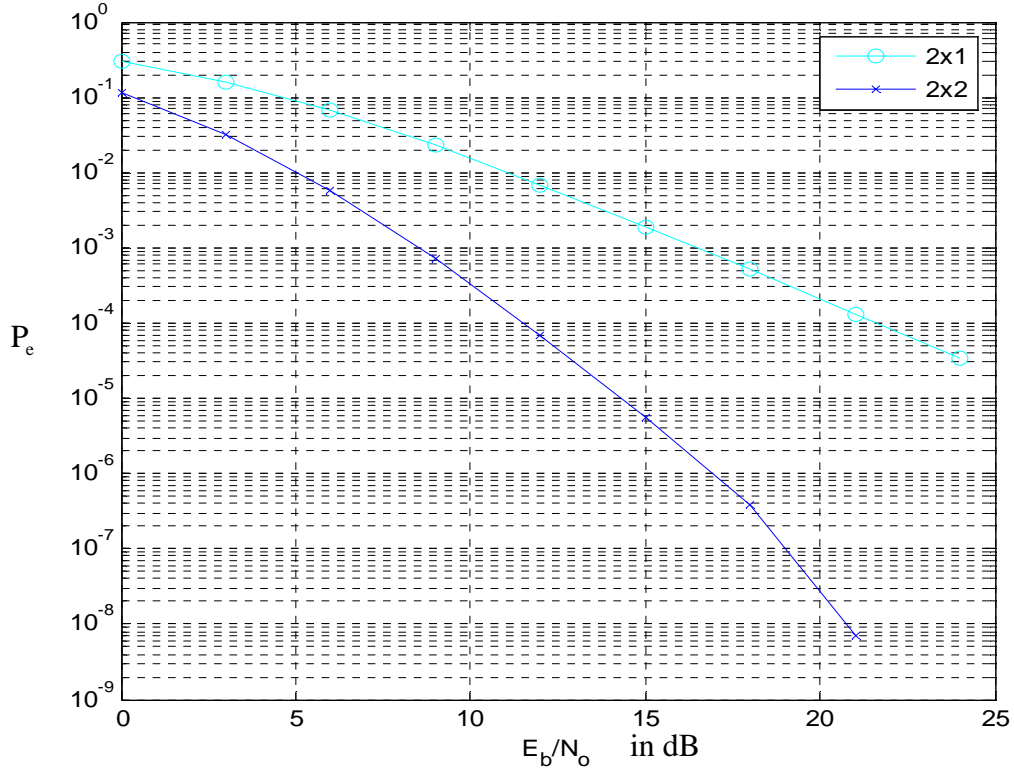


Figure 18. BER Performance of MIMO Systems over a Rayleigh Fading Channel (Two Transmitting Antennas.)

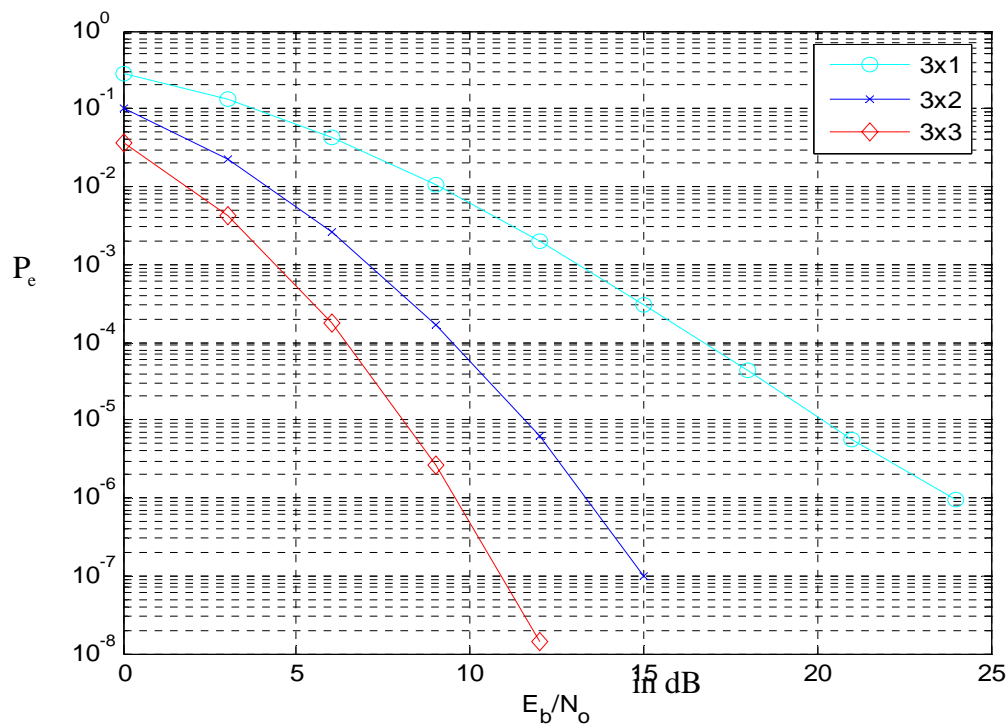


Figure 19. BER Performance of MIMO Systems over a Rayleigh Fading Channel (Three Transmitting Antennas.)

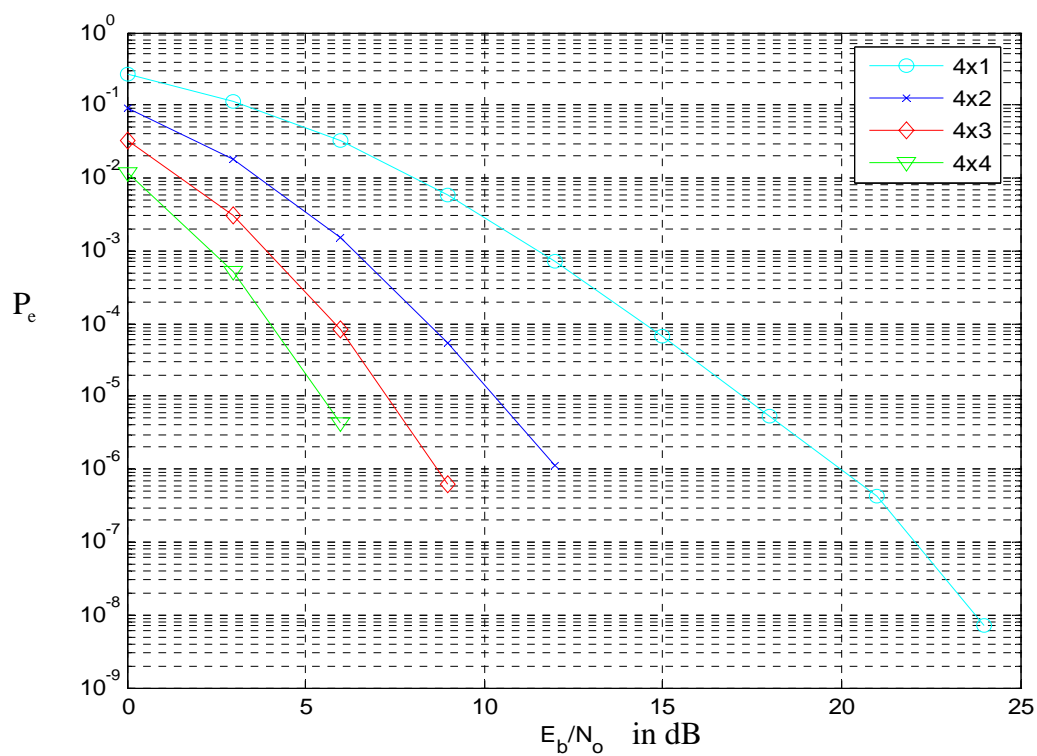


Figure 20. BER Performance of MIMO Systems over a Rayleigh Fading Channel (Four Transmitting Antennas.)

From the graphs depicted in Figures 18 to 20, Table 3 summarizes the energy per bit to noise power spectral density ratio required for each of the schemes to achieve a BER performance of  $10^{-6}$ .

MIMO Scheme	4×4	4×3	3×3	4×2	3×2	2×2	4×1	3×1
$E_b / N_0$ in dB	$\approx 7$	8.8	9.5	12	13	17.5	20	24

Table 3. Required  $E_b/N_0$  for a BER of  $10^{-6}$  for Different MIMO Schemes

From Figures 18 to 20 and Table 3, the  $E_b / N_0$  performance improves as the diversity gain is increased. This conforms to the theoretical results presented in Chapter II, where it was shown that the magnitude of the recovered signal is much greater for a MIMO system than for the corresponding MISO system. Again, the BER performance improvement diminishes as the number of receive antennas increases since the BER formula is not a linear function of the received signal magnitude. It is clear that the MIMO system enhances system performance in the multi-path environment without requiring additional bandwidth.

#### D. PERFORMANCE OF A MISO SYSTEM OVER A SUI CHANNEL

The SUI channel model, introduced in Chapter III, takes into consideration the effects of path loss, delay spread, Doppler spread, K-factor, and co-channel interference on signals that propagate from sender to receiver. A MATLAB model for the SUI channel that considers different combinations of terrain type, delay spread, Doppler spread, and K-factor is provided in [8]. We will use SUI Channel Model #2 for all simulations, to allow direct comparison of all results. The SUI model represents the channel information as a sequence of complex weights, which indicate the effects of multi-path, magnitude fading, and phase shift. Consequently, the received signals are determined by multiplication and time-shifting operations on the transmitted signal, (i.e., SUI channel acts as a filter function). The sampling frequency is chosen to be  $f_s = 22.857$  MHz to conform to the IEEE 802.16a standard. A rate  $\frac{1}{2}$ , constraint length 7, convolution encoder of polynomials (171,133) i.e., the decimal representation of generator polynomials [16] is chosen for the channel coding. (The same channel coding

will also be used in the simulations presented in the next section.) We present results for two different antenna types—directional and omnidirectional—so that our simulation results can be compared with those presented in the literature [11]. As before, QPSK is the modulation scheme used for the simulation. The detailed parameters of the simulation are listed in Table 4. The BER performance as a function of  $E_b/N_0$  is plotted in Figures 21 (directional antenna case) and 22 (omnidirectional antenna case).

<b>SUI channel model</b>	#2
<b># of transmitted information bits for each <math>E_b / N_o</math></b>	15,360,000
<b># of symbols block transmitted for each <math>E_b / N_o</math></b>	40,000
<b>Convolution Encoder Polynomials</b>	171,133
<b># of Monte Carlo runs</b>	12

Table 4. Parameters of a MISO System over a SUI Channel.

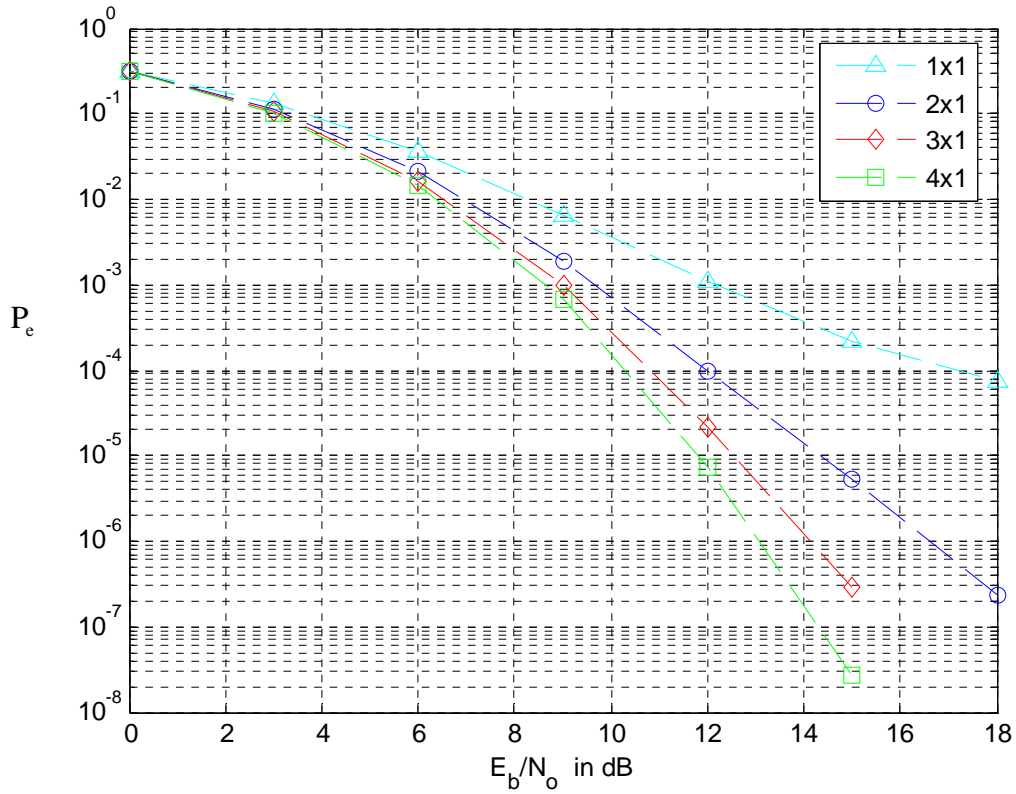


Figure 21. BER Performance of MISO Systems over a SUI Channel (Directional Antennas.)

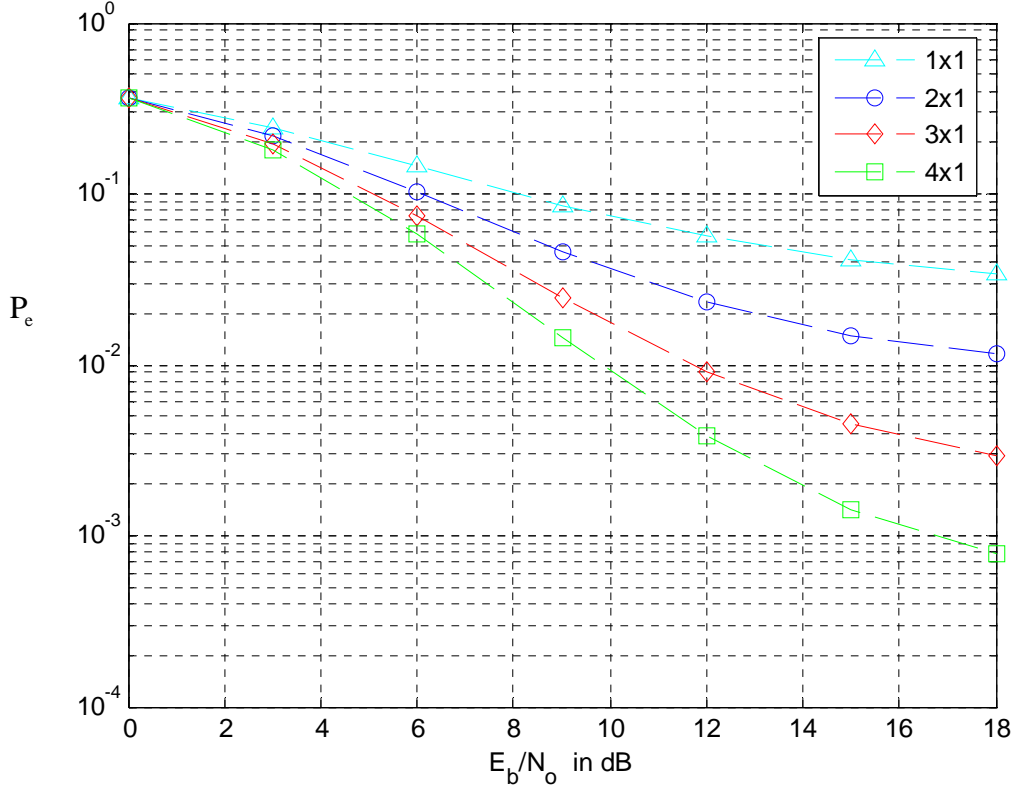


Figure 22. BER Performance of MISO Systems over a SUI Channel (Omni- Antennas.)

The performance shown in Figure 21 matches exactly with results presented in [11], which lends confidence to the correctness of our simulations. The BER performance improves as the number of antennas increases, as expected, and, comparing Figures 21 and 22, we note that the performance is better if directional antennas are used since a directional antenna has a greater effective area (i.e., a higher gain).

#### E. PERFORMANCE OF A MIMO SYSTEM OVER A SUI CHANNEL

We next evaluate the performance of MIMO systems over a SUI channel, using the same setting as the previous section, and assuming that all signal paths are uncorrelated. Table 4 lists simulation details.

Figures 23, 24 and 25 show the BER performance of MIMO systems over the SUI channel; the number of transmitting antennas is 2, 3 and 4, respectively. The results of Figure 23 agree with those presented in [11]. The BER performance trends are as expected: as the number of receive antennas was increased, the magnitude of the recovered signal increased (see Chapter II), and hence the BER performance improved for a fixed  $E_b/N_0$ . Less improvement is achieved with each additional receive antenna

since the BER is not a linear function of the magnitude of recovered signal. Comparing Figures in this section (Figure 23 to 25) to those in section C (Figure 18 to 20), the performances are improved due to the usage of directional antennas and channel coding.

The complexity of a MIMO system is much greater than the corresponding MISO system. While the transmitting sides of both systems are the same, intensive computation must be performed at the receiving side of the MIMO system. For example, the receiver in the 4×4 system must wait until the end of the complete transmission block before starting to decode.

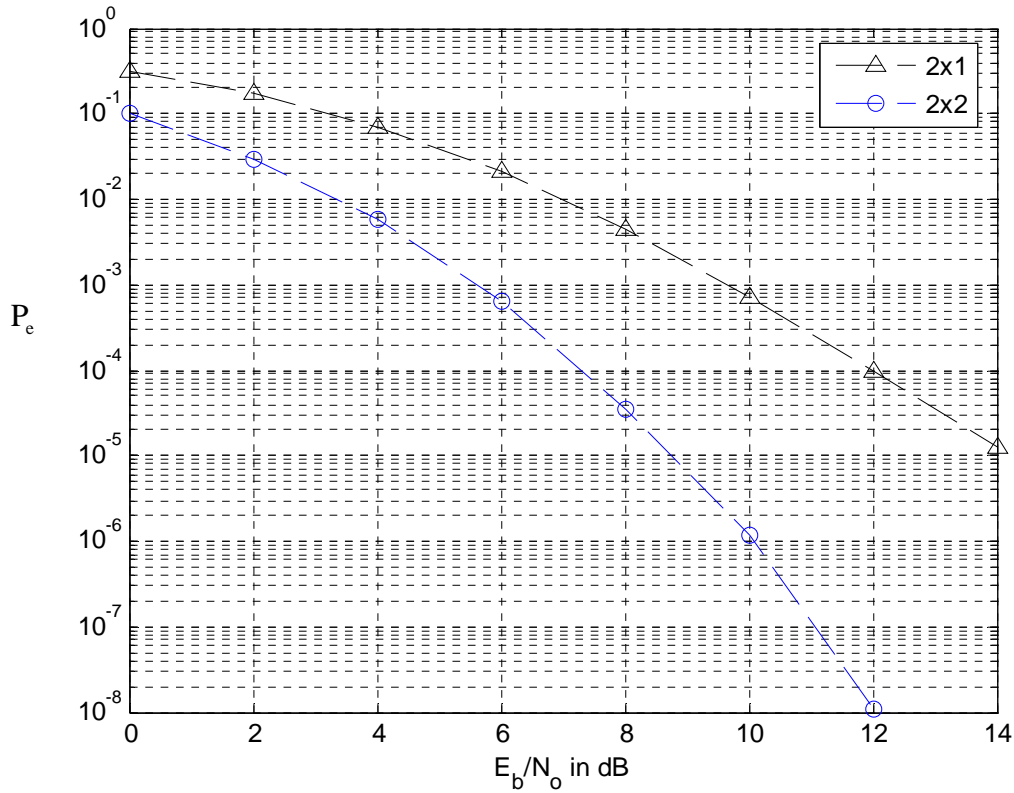


Figure 23. BER Performance of Two-Transmitter MIMO Systems over a SUI Channel (Two Directional Transmitting Antennas.)

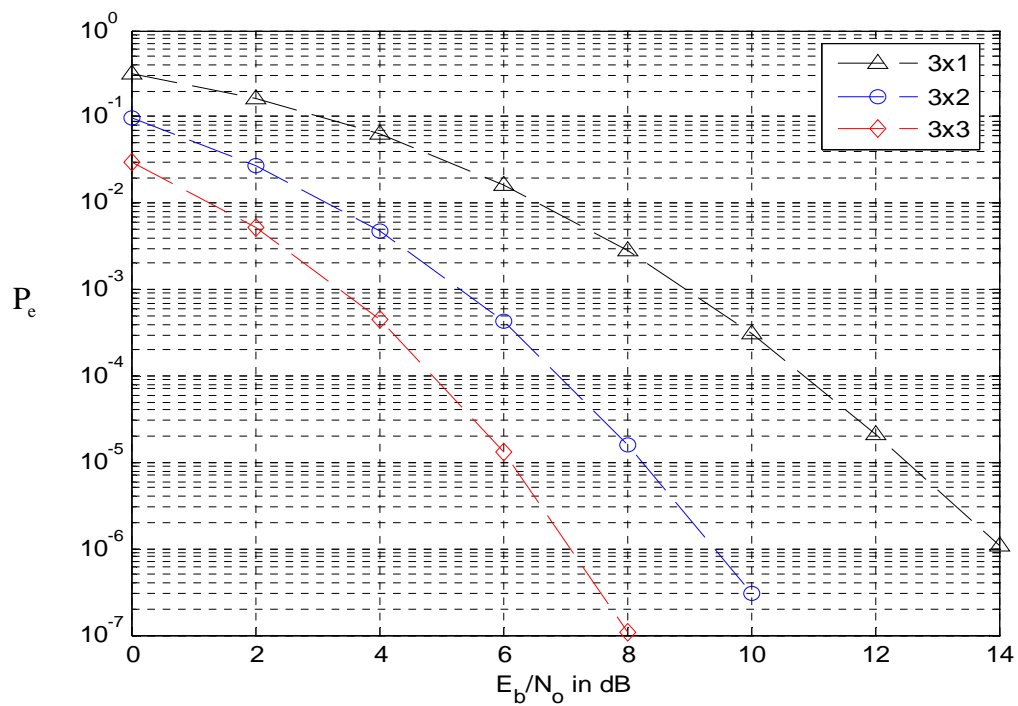


Figure 24. BER Performance of Three-Transmitter MIMO Systems over a SUI Channel (Three Directional Transmitting Antennas.)

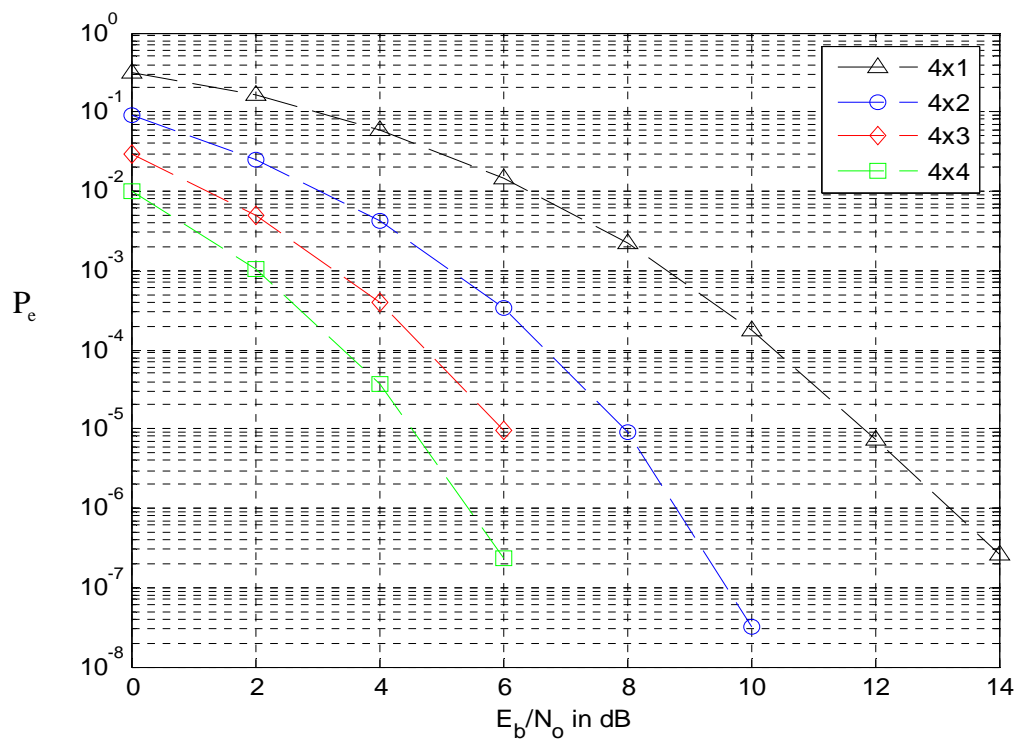


Figure 25. BER Performance of Four-Transmitter MIMO Systems over a SUI Channel (Four Directional Transmitting Antennas.)

## F. CHANNEL CORRELATION FOR THE SUI CHANNEL SCENARIO

In this section, the effects of channel correlation on system performance will be evaluated. As described in Chapter III, the diversity gain and the channel capacity will be significantly reduced if the channels are correlated. The technique for building a correlated SUI channel with specific correlation coefficients was presented in Chapter 3 (see Figure 6). Using the same simulation parameters as in the previous section (see Table 5), the BER performance of a particular MISO system will be evaluated for different values of the channel correlation coefficient.

The  $4 \times 1$  MISO scheme and the SUI 2 channel are used, and the correlation coefficient takes on values of 0 (no correlation), 0.1, 0.2, 0.4, 0.6, 0.8, and 1. A convolution encoder with polynomials (171,133) is used, and directional antennas are applied. The sampling frequency of the SUI channel is  $f_s = 22.857$  MHz in order to conform to the IEEE.16a standard. The simulation results are presented in Figure 26.

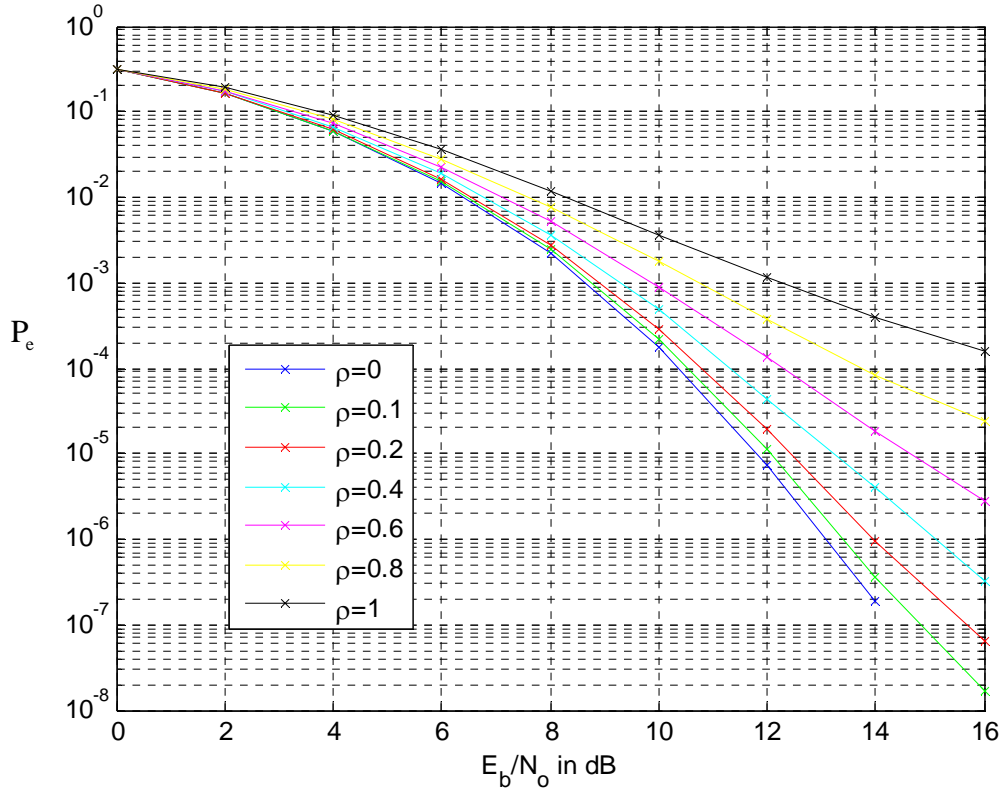


Figure 26. BER Performance Comparison of a  $4 \times 1$  MISO System over a SUI-2 Channel with Correlation Coefficient Variation (Four Directional Antennas.)



From Figure 26, the BER plot of zero-correlation exactly matches the same plot of Figure 21 (except less than  $10^{-6}$ ) and the correlation between channels degrades the performance significantly. The performance of 4×1 MISO scheme with correlation coefficient  $\rho = 1$  behaves similarly to that of 1×1 system (see Figure 21). This indicates that it is critical that a system designer has to investigate the potential causes of channel correlation and attempt to reduce these causes in order to ensure the designed system operates as expected.

#### **G. VARIATION IN SPACING BETWEEN ANTENNAS FOR THE SUI CHANNEL SCENARIO**

In some sensor network scenarios, the exact positions of the sensor nodes will be unknown, and, consequently, the spacing between any two sensor nodes must be modeled as a random variable. Each individual sensor node contains a single omnidirectional antenna (as well as a transceiver, a CPU and one or more sensors). If several sensor nodes cooperate, they can (at least in theory) operate as a MIMO system. Specifically, several sensor nodes (i.e., several antenna elements) might coordinate their transmissions, and essentially operate as the transmission end of a MISO system. Since sensor nodes are small, they carry small batteries and are thus very power-limited. Thus, if sensor nodes are collectively used as a MISO transmitter, some nodes that participate will quickly exhaust their batteries and the MISO transmission task will need to be passed off to another set of sensor nodes as time progresses. As the task of serving as a MISO transmitter is passed from one group of nodes to another, the actual geographic position of the MISO transmitter is translated to another location on the ground. It is simpler, however, to consider the MISO transmitter as not moving, but instead, being composed of antenna elements that move about with respect to each other. In other words, in this latter view, the MISO transmitter will be composed of a fixed number of antenna elements in a linear antenna configuration, with each pairs of nodes separated by a position that is modeled as a random variable. We will use the theoretical results of Chapter III to evaluate the effect that changes in element spacing has on performance.

Using a transmission frequency of  $f_c = 2.4$  GHz (the IEEE.11b/g standard), the signal wavelength is  $\lambda = 12.5$  cm. For the simulations presented in this section, we will apply the following assumptions:

1. The sensor nodes forming the MISO transmitter will consist of a uniform linear antenna array, i.e., the antennas will be arranged on a straight line, and the spacing between any two adjacent antennas will be equal to a uniform random variable  $d$ . (Although  $d$  is a random variable, for any specific instant, the same separation applies across the array.) The receiver will consist of a single omnidirectional antenna.
2. We assume that the correlation coefficients can be calculated by considering conditions at the transmitters. Note that a formula is available for calculating the correlation at the receiver based on parameters at the receivers (see Section 3.D). We assume that this same formula can be applied at the transmitters.
3. The distribution of angle of departure is modeled by a Laplacian distribution (see Equation (3.10) with standard deviation  $\sigma_A = 5^\circ$ ).

With these assumptions, the magnitude of the correlation coefficient as a function of the antenna spacing is as shown in Figure 14.

We consider that the transmit antenna spacing will uniformly vary between 12.5 cm and 37.5 cm (i.e., between  $\lambda$  and  $3\lambda$ , where  $\lambda$  is the carrier wavelength) and will not fall outside this range. A truncated curve of  $\theta = 0^\circ$  in Figure 14 is presented in Figure 27, showing only the antenna spacings of interest. From Figure 27, the correlation coefficient decreases almost linearly from 0.8704 to 0.4248. Thus, the correlation coefficient is an uniform random variable that will vary in this range (0.8704 to 0.4248). From a performance perspective, the best (i.e., lowest) correlation coefficient (0.4248) occurs when the inter-antenna spacing is maximum (37.5 cm), and the worst correlation coefficient occurs when the inter-antenna spacing is minimal (12.5 cm). The probability density function describing the correlation coefficient will depend on the probability density function describing the spacing between antenna elements.

We will assume in our simulations that the MISO transmitter is shifted to a different set of sensor nodes every 10 blocks where each block consists of 384 symbols.

In other words, the value of the correlation coefficient will change every 3,840 symbol periods.

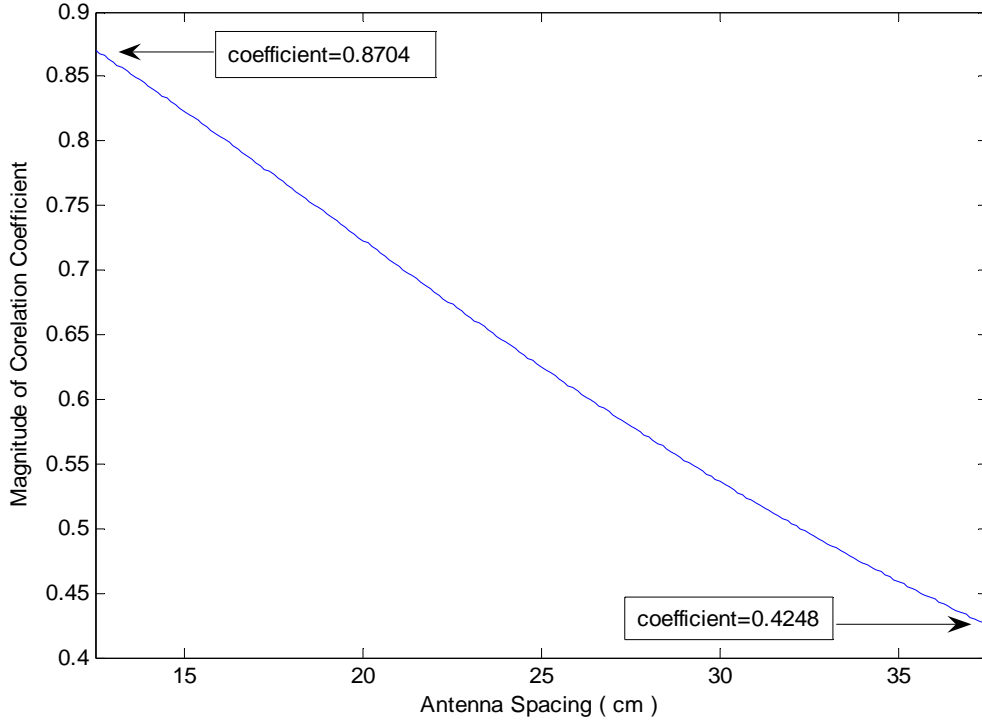


Figure 27. Truncated Curve of the Magnitude of the Correlation Coefficient versus Antenna Spacing ( $\theta = 0^\circ$ , Laplacian AOA with  $\sigma_A = 5^\circ$ .)

Against this backdrop, the BER performance as a function of  $E_b/N_0$  is shown in Figures 28, 29 and 30, for a  $2 \times 1$ ,  $3 \times 1$  and  $4 \times 1$  system, respectively.

For each of these figures, the upper-bound indicates the worst-case performance of smallest antenna separation and the lower-bound indicates the best-case performance of largest antenna spacing. For the upper and lower bound curves, the spacing is held constant. For the curve marked “Spacing Variation”, the spacing between elements changes every 10 space-time blocks where the correlation coefficient is a random variable drawn from a uniform distribution over the range of 0.8704 to 0.4248.

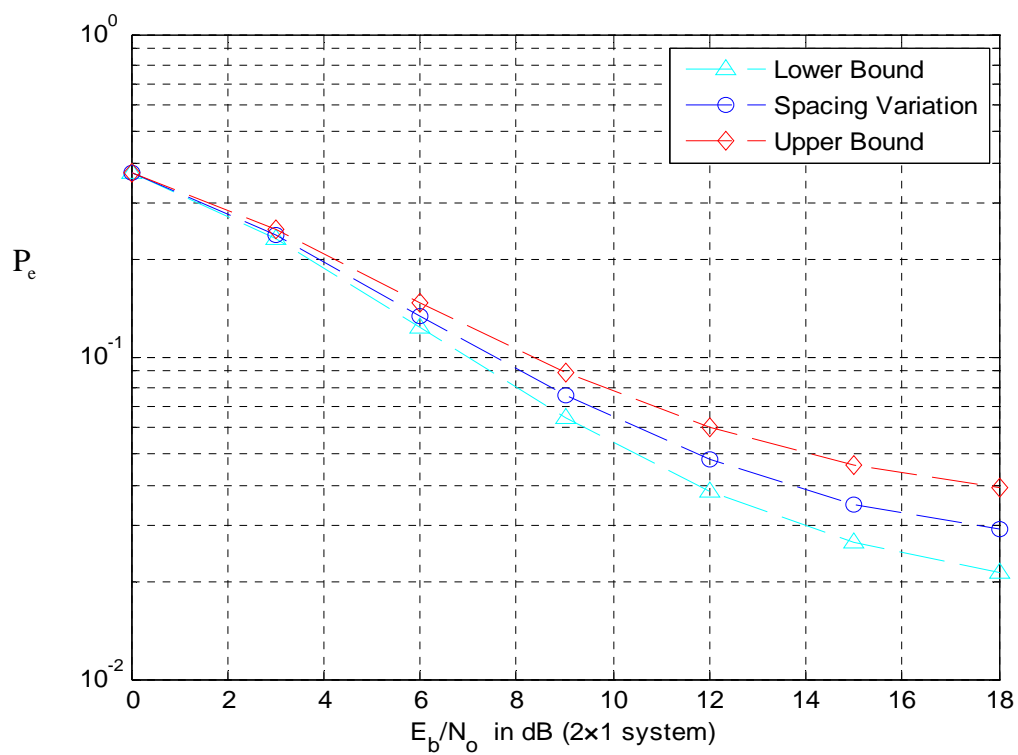


Figure 28. Performance with Spacing Variation (2×1 system.)

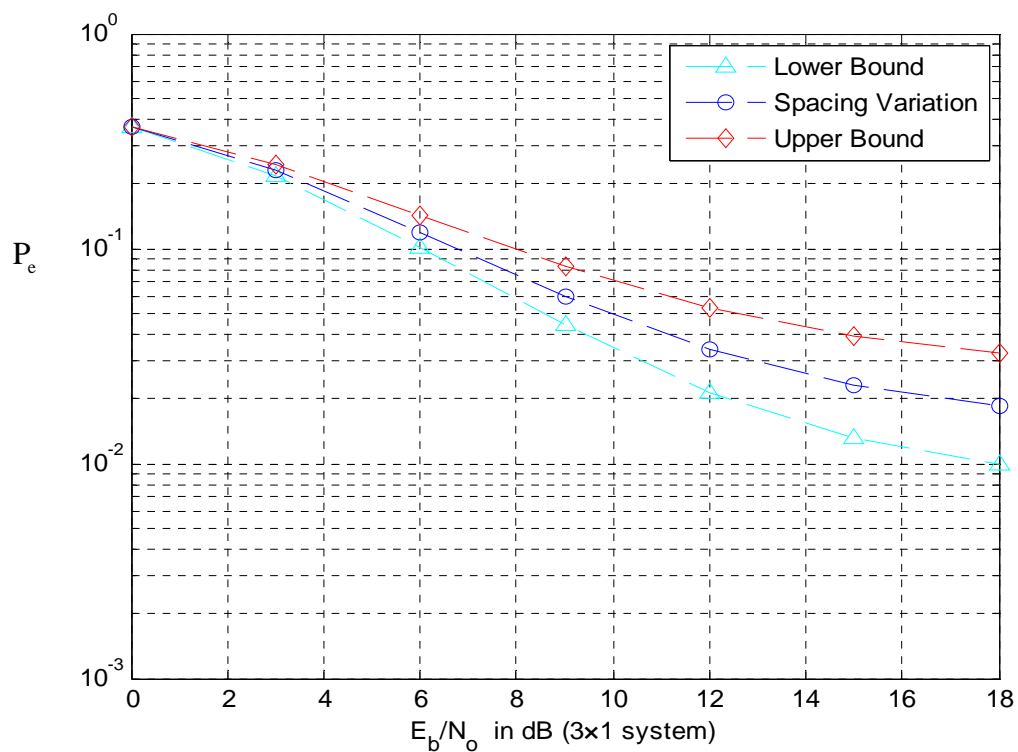


Figure 29. Performance with Spacing Variation (3×1 system.)

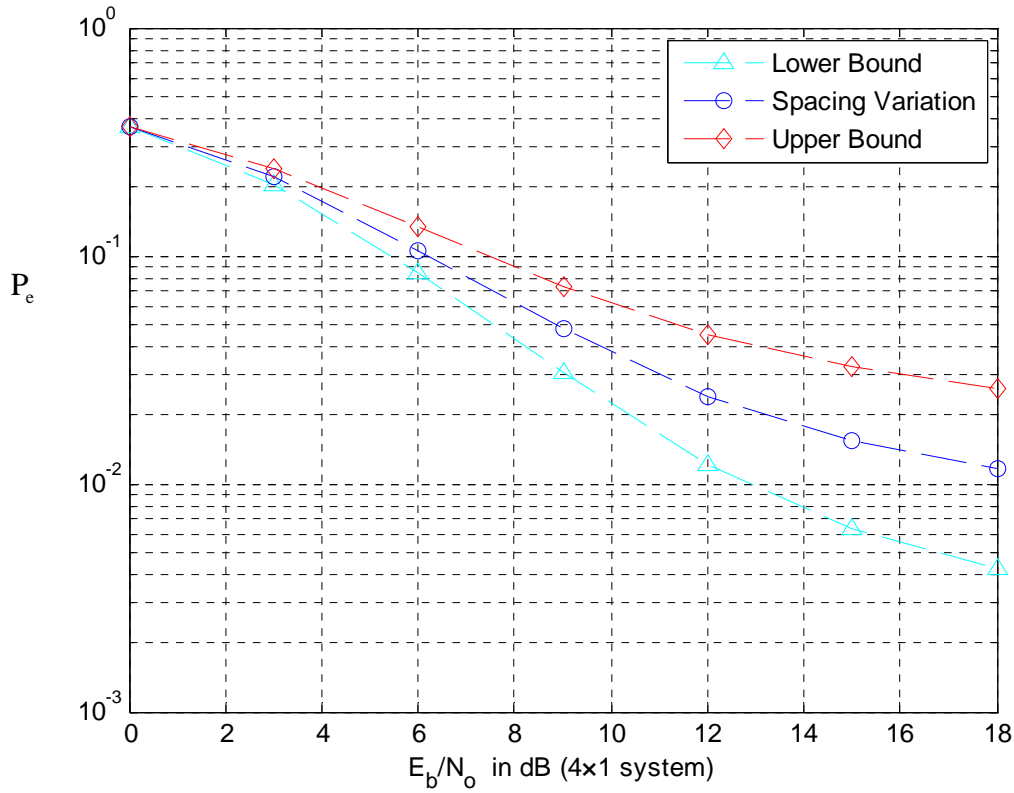


Figure 30. Performance with Spacing Variation (4×1 system.)

In Figures 28-30, the spacing variation curve results falls roughly in the middle between the upper and lower bounds, as expected. The results reported here on antenna spacing are preliminary. As further research, the simulation could be extended to consider arbitrary antenna element configurations.

In this chapter, several different scenarios are simulated in terms of BER versus  $E_b/N_o$  and simulation results included. The first two simulations are based on Rayleigh fading without channel memory. The results showed that different antenna combinations in the transmitting and receiving sides caused different performance. The third and fourth simulations were based on the SUI channel and took the channel's memory into account. The fifth simulation showed that correlation caused the performance to degrade significantly. Finally, the effects of the variation of elements spacing on performance was presented.

THIS PAGE INTENTIONALLY LEFT BLANK

## V. CONCLUSION

This thesis investigated the performance of MIMO systems that use STBC. Such systems can be employed to improve the BER performance of wireless communication systems and can counter the detrimental effects of channel fading and other distortion phenomena. A variety of different MIMO schemes were evaluated in simulation using the MATLAB programming language. MIMO systems can also be used to counter the performance degradation that results from correlation between multipath channels at a receiver. This thesis analyzed the relationship between channel correlation coefficients and antenna spacing through simulation for the scenario of a linear antenna configuration.

### A. SIGNIFICANT RESULTS

This thesis proposed a systematic method for designing a MIMO system with an arbitrary number of transmit and receive antennas. Simulation results were used to evaluate the performance of MIMO-STBC systems designed using our approach, in both Rayleigh fading channels and in channels based on actual measurements (the Stanford University Interim channel model). We presented a formula that can be readily used to determine the rate (i.e., the ratio of the number of symbols transmitted over the number of symbol intervals required) of systems that employ our design. Three criteria—the rate of the scheme, the BER and the complexity—were used to analyze our scheme against alternative MIMO designs available in the literature, and we described the trade-offs that a designer must consider when contemplating the choice of a MIMO system in a specific application.

This thesis studied the relationship between channel correlation and antenna spacing for the case of a uniform linear antenna configuration. Through analysis and simulation, we were able to analyze the effects of channel correlation on the performance of MIMO systems. We noted that MIMO systems can employ the multipath phenomenon to reduce the detrimental effects of channel correlation, and that the greater the number of multi-path arrivals, the less correlated the channels will be. Thus, a MIMO system can take advantage of the large number of multi-path arrivals to achieve better performance.

## **B. SUGGESTIONS FOR FUTURE WORK**

One of the shortcomings of our proposed MIMO design scheme is that the rate decreases as the number of transmitting antennas increases. As a consequence of the lower rate, the system suffers more delay. Further investigation of systematic MIMO design schemes might be aimed at achieving higher rates than reported here while attempting to match the BER performance achieved in this work.

A number of our simulation studies employed directional antennas, and, as expected, these systems showed better performance than similar systems that used omnidirectional antennas. The use of directional antennas in different MIMO application scenarios is worthy of further research.

Our analytical and simulation results assumed that the receiver has accurate CSI. Obviously, perfect CSI may not be available all the time, particularly if the channel exhibits dramatic fading, and using outdated CSI information might result in poor performance. The traditional way to measure the channel state is by employing pilot sequences. Further research might examine how often to transmit pilot sequences and how long such sequences should be, in the face of channel variations. Such research would necessarily consider the performance degradation due to imperfect CSI.

Further research would explore the use of different error correcting channel codes. In this thesis, a convolutional code was used for several simulations. It would be interesting to explore whether different channel coding schemes might better exploit the characteristics of the MIMO system structure, leading to better performance.

In the literature, several probability distributions for the incoming signal AOA are thoroughly studied; only two of them are discussed in this thesis. The distribution of the AOA plays a critical role in determining the correlation between channels; this thesis showed that the BER performance curves differ dramatically as a result of different AOA distributions. The configuration of antenna elements and antenna spacing also determine the extent of channel correlation. The scope of our studies was restricted to linear antenna arrays. Further research may consider generalizing our results to arbitrary antenna positioning, and different AOA distributions.



## APPENDIX

This appendix briefly explains the Matlab code used for simulation, the structure of each simulation function, and some important sub-functions that apply to several simulations.

### A. STRUCTURE OF SIMULATION FILES

The simulation program used to generate the results presented in Chapter IV is based on nested iterations that are controlled by three parameters. The main (outer) iteration is set by the parameter “MonteCarlo,” which determines how many Monte Carlo iterations will be used for the simulation. Typically, twelve Monte Carlo iterations are chosen for simulations presented in Chapter IV. The second parameter controls the number of runs for each value of  $E_b/N_o$  and is denoted “EbNo\_db”. Based on this parameter, the simulation program calculates the number of error symbols at each  $E_b/N_o$  point for each Monte Carlo run, and generates a plot showing the average BER for this point. The third user-input parameter, “n,” defines how many bit iterations will be conducted at each specific  $E_b/N_o$  point for each Monte Carlo run. A sequence of bit streams will be generated in an iteration, mapped into baseband symbols for transmission, encoded by space-time coding and/or error correction coding, sent over a simulated channel model, and received and decoded at receiver where the number of error bits is calculated and used for determining the BER. Figure 31 shows a conceptual diagram illustrating the roles of these three parameters.

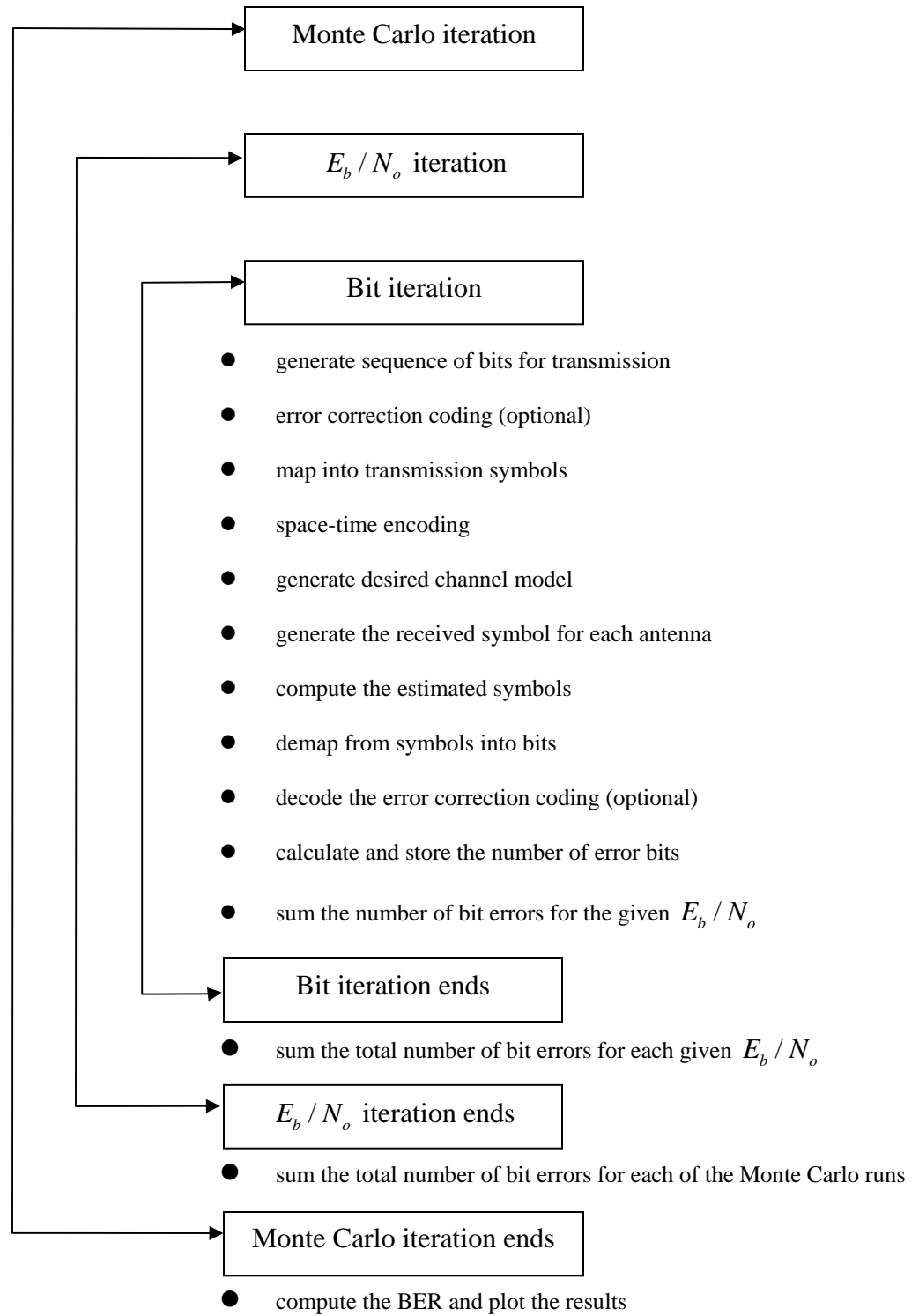


Figure 31. A Conceptual Diagram of Simulation Structure (After Reference [11])

## B. EXPLANATION OF THE SUB-FUNCTIONS

This section briefly explains some important sub-functions that apply to several simulations. Some sub-functions are taken directly from [11], some are modified, and some are new in this thesis. The maximum number of transmitting antennas is limited to four in this thesis, although this number can be increased.

*add\_noise2sequ.m* adds complex Gaussian distributed random noise of zero mean and desired noise variance to the input sequence.

*bin\_2\_mary.m* converts the convolutionally-encoded bits to  $m$ -ary symbols. The result of this function is a vector that indicates a specific point in  $m$ -ary modulation constellation.

*pskmod.m* is the Matlab's built-in function, which generates a complex envelope according to the inputs specifying the type of phase shift keying modulation and the point in the constellation.

*STBC\_encode\_two.m* takes a group of 384 symbols as a block of space-time coding and generates the corresponding coded sequences for two transmitting antennas.

*STBC\_encode\_three.m* takes a group of 384 symbols as a block of space-time coding and generates the corresponding coded sequences for three transmitting antennas.

*STBC\_encode\_four.m* takes a group of 384 symbols as a block of space-time coding and generates the corresponding coded sequences for four transmitting antennas.

*SUI\_model.m* generates a  $4 \times 1$  complex vector of overall channel information, fading component, fixed component, and normalization factor according to the specified SUI index channel parameters. The directivity of the antenna can be chosen (omni or  $30^\circ$  directional antenna).

*SUI\_model\_resample.m* samples the channel information of *SUI\_model.m* with the desired stored delay spread values for all six SUI channels and sampling frequency.

*non\_causal\_filter.m* operates the decoding process when a non-causal filter is needed.

*pskdemod.m* is the Matlab's built-in function, which demaps the estimated complex sequence back to decimal numbers according to the specified type of phase shift keying modulation constellation.

*mary\_2\_bin.m* converts the decimal number to bits by using the specified type of  $m$ -ary constellation.

*MIMO\_encode.m* takes every 12 symbols in the input sequence into MIMO encoding and generates the outputs containing four different MIMO encoded sequences (one to four transmitting antennas.)

### C. MATLAB CODES

```
% Title      : This function simulates MIMO over Rayleigh fading channels:
%            Mx1,Mx2, until MxM, MIMO over PSK modulation scheme (M<=4)
%            No channel encoder is used here.
%            BER curves of all schemes are presented in one figure
%            One tap, zero mean , circularly symmetric, complex
%            channels are created for each transmitted symbol
% Author     : Nieh, Jo-Yen, Naval Postgraduate School, January 2006
%-----
% Input
% m : the number of transmitting antenna
% Output
% bit error rate of Mx1,Mx2, until MxM in one plot
%-----
% Simulation parameters
clear all;clc;close all
m=4;
M=4; % m-ary constellation, QPSK = 4, 16QAM = 16
n=log2(M)*12*100; % total number of bits to be transmitted for the given SNR value
MonteCarlo = 1; % number of runs to be simulated
EbNo_db = 0:5:30; % SNR interval to be simulated

%-----
% Noise parameters
sig = sqrt(0.5); % noise std
```

```

No = 2*sig^2;
No_db = 10*log10(No);

%-----

awgn = 0; % to see AWGN performance awgn = 1;

% creating 'total bit errors array' for all Monte Carlo runs

total_bit_errors_monte_1 = []; total_bit_errors_monte_2 = []; total_bit_errors_monte_3 = [];
total_bit_errors_monte_4 = [];

for run = 1:MonteCarlo % Monte Carlo iteration

    % resetting 'total bit errors array' for all SNR values
    total_bit_errors_1 = []; total_bit_errors_2 = []; total_bit_errors_3 = []; total_bit_errors_4 = [];

    for k=1:length(EbNo_db); % SNR iteration

        % resetting 'number of errors' for the given SNR value
        bit_errors_1 = 0; bit_errors_2 = 0; bit_errors_3 = 0; bit_errors_4 = 0;

        disp(['SNR = ',num2str(EbNo_db(k)), ' Monte Carlo run = ',num2str(run)]);

        for block=1:12:n; % bit iteration (one QPSK symbol of 2 bits)
            number_of_bit_errors_1 = 0; number_of_bit_errors_2 = 0;
            number_of_bit_errors_3 = 0; number_of_bit_errors_4 = 0;

            % generating the transmitted symbols( six symbols per iteration )
            x = randint(1,12*log2(M));
            x_decimal = bin_2_mary(log2(M),x); % binary to m-ary conversion
            x_mapped = pskmod(x_decimal,M);

            % MIMO encoding

```

```

[X1, X2, X3, X4] = MIMO_encode(x_mapped); % X1:x_mapped(1x12); X2:alamouti
encoding(2x12);

% X3:Three antenna(3x16); X4:four(4x21)

if m==1
    X=X1;
elseif m==2
    X=X2;
elseif m==3 % determining transmitting signals ( m by ? )
    X=X3;
elseif m==4
    X=X4;
end

% creating the channel( assuming CSI is stable during all the transmissions)
if awgn == 0;
    h1 = MIMO_channel(1,m);
    h2 = MIMO_channel(2,m); % ie. h1=1xm; h2=2xm; h3=3xm; h4=4xm
    h3 = MIMO_channel(3,m);
    h4 = MIMO_channel(4,m);
else
    h1 = ones(1,m); h2 = ones(2,m);
    h3 = ones(3,m); h4 = ones(4,m);
end

% creating 'total bit errors array' for all Monte Carlo runs
EsNo_db = EbNo_db(k) + 10*log10(log2(M)); % Es/No_db = Eb/No_db + q_db ; from
Eb/No_db and q to Es/No_db;
Es_db = EsNo_db + No_db;
Es = 10.^(Es_db/10); % get the mag of Es

for i=1:m
    if i==1 % mx1 received signals
        R_1 = add_noise2sequ(sqrt(Es)*h1*X,1,sig);
    end

    if i==2 % mx2 received signals

```

```

    R_2 = add_noise2sequ(sqrt(Es)*h2*X,2,sig);
end

if i==3 % mx3 received signals
    R_3 = add_noise2sequ(sqrt(Es)*h3*X,3,sig);
end

if i==4 % mx4 received signals
    R_4 = add_noise2sequ(sqrt(Es)*h4*X,4,sig);
end

% computing the estimates

if m==1 % m=1 case
    S1_est = conj(h1)*R_1;
end

if m==2 % m=2 case
    S1_est = [];
    for index=1:2:length(R_1)
        E1 = conj(h1(1))*R_1(index) + h1(2)*conj(R_1(index+1));
        E2 = conj(h1(2))*R_1(index) - h1(1)*conj(R_1(index+1));
        S1_est=[S1_est E1 E2];
    end
    S2_est = [];
    for index=1:2:length(R_2)
        E1 = conj(h2(1,1))*R_2(1,index) + h2(1,2)*conj(R_2(1,index+1)) +
        conj(h2(2,1))*R_2(2,index) + h2(2,2)*conj(R_2(2,index+1));
        E2 = conj(h2(1,2))*R_2(1,index) - h2(1,1)*conj(R_2(1,index+1)) +
        conj(h2(2,2))*R_2(2,index) - h2(2,1)*conj(R_2(2,index+1));
        S2_est=[S2_est E1 E2];
    end
end
end

```

```

if m==3      % m=3 case
    S1_est = [];
    for index=1:4:length(R_1)
        E1 = conj(h1(1))*R_1(index) + h1(2)*conj(R_1(index+1)) + h1(3)*conj(R_1(index+2));
        E2 = conj(h1(2))*R_1(index) - h1(1)*conj(R_1(index+1)) + h1(3)*conj(R_1(index+3));
        E3 = conj(h1(3))*R_1(index) - h1(1)*conj(R_1(index+2)) - h1(2)*conj(R_1(index+3));
        S1_est=[S1_est E1 E2 E3];
    end
    S2_est = [];
    for index=1:4:length(R_2)
        E1 = conj(h2(1,1))*R_2(1,index) + h2(1,2)*conj(R_2(1,index+1)) +
h2(1,3)*conj(R_2(1,index+2)) + ...
        conj(h2(2,1))*R_2(2,index) + h2(2,2)*conj(R_2(2,index+1)) +
h2(2,3)*conj(R_2(2,index+2)) ;

        E2 = conj(h2(1,2))*R_2(1,index) - h2(1,1)*conj(R_2(1,index+1)) +
h2(1,3)*conj(R_2(1,index+3)) + ...
        conj(h2(2,2))*R_2(2,index) - h2(2,1)*conj(R_2(2,index+1)) +
h2(2,3)*conj(R_2(2,index+3)) ;

        E3 = conj(h2(1,3))*R_2(1,index) - h2(1,1)*conj(R_2(1,index+2)) -
h2(1,2)*conj(R_2(1,index+3)) + ...
        conj(h2(2,3))*R_2(2,index) - h2(2,1)*conj(R_2(2,index+2)) -
h2(2,2)*conj(R_2(2,index+3)) ;

        S2_est=[S2_est E1 E2 E3];
    end
    S3_est = [];
    for index=1:4:length(R_3)
        E1 = conj(h3(1,1))*R_3(1,index) + h3(1,2)*conj(R_3(1,index+1)) +
h3(1,3)*conj(R_3(1,index+2)) + ...
        conj(h3(2,1))*R_3(2,index) + h3(2,2)*conj(R_3(2,index+1)) +
h3(2,3)*conj(R_3(2,index+2)) + ...
        conj(h3(3,1))*R_3(3,index) + h3(3,2)*conj(R_3(3,index+1)) +
h3(3,3)*conj(R_3(3,index+2)) ;

```



```

E2 = conj(h3(1,2))*R_3(1,index) - h3(1,1)*conj(R_3(1,index+1)) +
h3(1,3)*conj(R_3(1,index+3)) + ...
      conj(h3(2,2))*R_3(2,index) - h3(2,1)*conj(R_3(2,index+1)) +
h3(2,3)*conj(R_3(2,index+3)) + ...
      conj(h3(3,2))*R_3(3,index) - h3(3,1)*conj(R_3(3,index+1)) +
h3(3,3)*conj(R_3(3,index+3)) ;

```

```

E3 = conj(h3(1,3))*R_3(1,index) - h3(1,1)*conj(R_3(1,index+2)) -
h3(1,2)*conj(R_3(1,index+3)) + ...
      conj(h3(2,3))*R_3(2,index) - h3(2,1)*conj(R_3(2,index+2)) -
h3(2,2)*conj(R_3(2,index+3)) + ...
      conj(h3(3,3))*R_3(3,index) - h3(3,1)*conj(R_3(3,index+2)) -
h3(3,2)*conj(R_3(3,index+3)) ;

```

```

S3_est=[S3_est E1 E2 E3];
end
end

```

```

if m==4      % m=4 case
S1_est = [];
for index=1:7:length(R_1)
E1 = conj(h1(1))*R_1(index) + h1(2)*conj(R_1(index+1)) + h1(3)*conj(R_1(index+2)) +
h1(4)*conj(R_1(index+3));
E2 = conj(h1(2))*R_1(index) - h1(1)*conj(R_1(index+1)) + h1(3)*conj(R_1(index+4)) +
h1(4)*conj(R_1(index+5));
E3 = conj(h1(3))*R_1(index) - h1(1)*conj(R_1(index+2)) - h1(2)*conj(R_1(index+4)) +
h1(4)*conj(R_1(index+6));
E4 = conj(h1(4))*R_1(index) - h1(1)*conj(R_1(index+3)) - h1(2)*conj(R_1(index+5)) -
h1(3)*conj(R_1(index+6));
S1_est=[S1_est E1 E2 E3 E4];
end
S2_est = [];
for index=1:7:length(R_2)
E1 = conj(h2(1,1))*R_2(1,index) + h2(1,2)*conj(R_2(1,index+1)) +
h2(1,3)*conj(R_2(1,index+2)) + h2(1,4)*conj(R_2(1,index+3)) + ...
      conj(h2(2,1))*R_2(2,index) + h2(2,2)*conj(R_2(2,index+1)) +

```

$h2(2,3)*conj(R\_2(2,index+2)) + h2(2,4)*conj(R\_2(2,index+3))$  ;

$E2 = conj(h2(1,2))*R\_2(1,index) - h2(1,1)*conj(R\_2(1,index+1)) +$   
 $h2(1,3)*conj(R\_2(1,index+4)) + h2(1,4)*conj(R\_2(1,index+5)) + \dots$   
 $conj(h2(2,2))*R\_2(2,index) - h2(2,1)*conj(R\_2(2,index+1)) +$   
 $h2(2,3)*conj(R\_2(2,index+4)) + h2(2,4)*conj(R\_2(2,index+5))$  ;

$E3 = conj(h2(1,3))*R\_2(1,index) - h2(1,1)*conj(R\_2(1,index+2)) -$   
 $h2(1,2)*conj(R\_2(1,index+4)) + h2(1,4)*conj(R\_2(1,index+6)) + \dots$   
 $conj(h2(2,3))*R\_2(2,index) - h2(2,1)*conj(R\_2(2,index+2)) -$   
 $h2(2,2)*conj(R\_2(2,index+4)) + h2(2,4)*conj(R\_2(2,index+6))$  ;

$E4 = conj(h2(1,4))*R\_2(1,index) - h2(1,1)*conj(R\_2(1,index+3)) -$   
 $h2(1,2)*conj(R\_2(1,index+5)) - h2(1,3)*conj(R\_2(1,index+6)) + \dots$   
 $conj(h2(2,4))*R\_2(2,index) - h2(2,1)*conj(R\_2(2,index+3)) -$   
 $h2(2,2)*conj(R\_2(2,index+5)) - h2(2,3)*conj(R\_2(2,index+6))$  ;

$S2\_est=[S2\_est\ E1\ E2\ E3\ E4];$

end

$S3\_est = [];$

for index=1:7:length(R\_3)

$E1 = conj(h3(1,1))*R\_3(1,index) + h3(1,2)*conj(R\_3(1,index+1)) +$   
 $h3(1,3)*conj(R\_3(1,index+2)) + h3(1,4)*conj(R\_3(1,index+3)) + \dots$   
 $conj(h3(2,1))*R\_3(2,index) + h3(2,2)*conj(R\_3(2,index+1)) +$   
 $h3(2,3)*conj(R\_3(2,index+2)) + h3(2,4)*conj(R\_3(2,index+3)) + \dots$   
 $conj(h3(3,1))*R\_3(3,index) + h3(3,2)*conj(R\_3(3,index+1)) +$   
 $h3(3,3)*conj(R\_3(3,index+2)) + h3(3,4)*conj(R\_3(3,index+3))$  ;

$E2 = conj(h3(1,2))*R\_3(1,index) - h3(1,1)*conj(R\_3(1,index+1)) +$   
 $h3(1,3)*conj(R\_3(1,index+4)) + h3(1,4)*conj(R\_3(1,index+5)) + \dots$   
 $conj(h3(2,2))*R\_3(2,index) - h3(2,1)*conj(R\_3(2,index+1)) +$   
 $h3(2,3)*conj(R\_3(2,index+4)) + h3(2,4)*conj(R\_3(2,index+5)) + \dots$   
 $conj(h3(3,2))*R\_3(3,index) - h3(3,1)*conj(R\_3(3,index+1)) +$   
 $h3(3,3)*conj(R\_3(3,index+4)) + h3(3,4)*conj(R\_3(3,index+5))$  ;

$E3 = conj(h3(1,3))*R\_3(1,index) - h3(1,1)*conj(R\_3(1,index+2)) -$   
 $h3(1,2)*conj(R\_3(1,index+4)) + h3(1,4)*conj(R\_3(1,index+6)) + \dots$

```

        conj(h3(2,3))*R_3(2,index) - h3(2,1)*conj(R_3(2,index+2)) -
h3(2,2)*conj(R_3(2,index+4)) + h3(2,4)*conj(R_3(2,index+6)) + ...
        conj(h3(3,3))*R_3(3,index) - h3(3,1)*conj(R_3(3,index+2)) -
h3(3,2)*conj(R_3(3,index+4)) + h3(3,4)*conj(R_3(3,index+6)) ;

E4 = conj(h3(1,4))*R_3(1,index) - h3(1,1)*conj(R_3(1,index+3)) -
h3(1,2)*conj(R_3(1,index+5)) - h3(1,3)*conj(R_3(1,index+6)) + ...
        conj(h3(2,4))*R_3(2,index) - h3(2,1)*conj(R_3(2,index+3)) -
h3(2,2)*conj(R_3(2,index+5)) - h3(2,3)*conj(R_3(2,index+6)) + ...
        conj(h3(3,4))*R_3(3,index) - h3(3,1)*conj(R_3(3,index+3)) -
h3(3,2)*conj(R_3(3,index+5)) - h3(3,3)*conj(R_3(3,index+6)) ;
S3_est=[S3_est E1 E2 E3 E4];
end
S4_est = [];
for index=1:length(R_4)
    E1 = conj(h4(1,1))*R_4(1,index) + h4(1,2)*conj(R_4(1,index+1)) +
h4(1,3)*conj(R_4(1,index+2)) + h4(1,4)*conj(R_4(1,index+3)) + ...
        conj(h4(2,1))*R_4(2,index) + h4(2,2)*conj(R_4(2,index+1)) +
h4(2,3)*conj(R_4(2,index+2)) + h4(2,4)*conj(R_4(2,index+3)) + ...
        conj(h4(3,1))*R_4(3,index) + h4(3,2)*conj(R_4(3,index+1)) +
h4(3,3)*conj(R_4(3,index+2)) + h4(3,4)*conj(R_4(3,index+3)) + ...
        conj(h4(4,1))*R_4(4,index) + h4(4,2)*conj(R_4(4,index+1)) +
h4(4,3)*conj(R_4(4,index+2)) + h4(4,4)*conj(R_4(4,index+3)) ;

    E2 = conj(h4(1,2))*R_4(1,index) - h4(1,1)*conj(R_4(1,index+1)) +
h4(1,3)*conj(R_4(1,index+4)) + h4(1,4)*conj(R_4(1,index+5)) + ...
        conj(h4(2,2))*R_4(2,index) - h4(2,1)*conj(R_4(2,index+1)) +
h4(2,3)*conj(R_4(2,index+4)) + h4(2,4)*conj(R_4(2,index+5)) + ...
        conj(h4(3,2))*R_4(3,index) - h4(3,1)*conj(R_4(3,index+1)) +
h4(3,3)*conj(R_4(3,index+4)) + h4(3,4)*conj(R_4(3,index+5)) + ...
        conj(h4(4,2))*R_4(4,index) - h4(4,1)*conj(R_4(4,index+1)) +
h4(4,3)*conj(R_4(4,index+4)) + h4(4,4)*conj(R_4(4,index+5)) ;

    E3 = conj(h4(1,3))*R_4(1,index) - h4(1,1)*conj(R_4(1,index+2)) -
h4(1,2)*conj(R_4(1,index+4)) + h4(1,4)*conj(R_4(1,index+6)) + ...
        conj(h4(2,3))*R_4(2,index) - h4(2,1)*conj(R_4(2,index+2)) -

```

$$\begin{aligned}
& h4(2,2)*conj(R\_4(2,index+4)) + h4(2,4)*conj(R\_4(2,index+6)) + \dots \\
& \quad conj(h4(3,3))*R\_4(3,index) - h4(3,1)*conj(R\_4(3,index+2)) - \\
& h4(3,2)*conj(R\_4(3,index+4)) + h4(3,4)*conj(R\_4(3,index+6)) + \dots \\
& \quad conj(h4(4,3))*R\_4(4,index) - h4(4,1)*conj(R\_4(4,index+2)) - \\
& h4(4,2)*conj(R\_4(4,index+4)) + h4(4,4)*conj(R\_4(4,index+6)) ;
\end{aligned}$$

$$\begin{aligned}
E4 = & conj(h4(1,4))*R\_4(1,index) - h4(1,1)*conj(R\_4(1,index+3)) - \\
& h4(1,2)*conj(R\_4(1,index+5)) - h4(1,3)*conj(R\_4(1,index+6)) + \dots \\
& \quad conj(h4(2,4))*R\_4(2,index) - h4(2,1)*conj(R\_4(2,index+3)) - \\
& h4(2,2)*conj(R\_4(2,index+5)) - h4(2,3)*conj(R\_4(2,index+6)) + \dots \\
& \quad conj(h4(3,4))*R\_4(3,index) - h4(3,1)*conj(R\_4(3,index+3)) - \\
& h4(3,2)*conj(R\_4(3,index+5)) - h4(3,3)*conj(R\_4(3,index+6)) + \dots \\
& \quad conj(h4(4,4))*R\_4(4,index) - h4(4,1)*conj(R\_4(4,index+3)) - \\
& h4(4,2)*conj(R\_4(4,index+5)) - h4(4,3)*conj(R\_4(4,index+6)) ;
\end{aligned}$$

```

S4_est=[S4_est E1 E2 E3 E4];
end
end

```

**% demapping, decoding and bit comparison for single carrier systems**

```

if m>0
S1_est_decimal = pskdemod(S1_est,M);
S1_est_bit = mary_2_bin(log2(M),S1_est_decimal);
number_of_bit_errors_1 = sum(xor(S1_est_bit, x));
end

```

```

if m>1
S2_est_decimal = pskdemod(S2_est,M);
S2_est_bit = mary_2_bin(log2(M),S2_est_decimal);
number_of_bit_errors_2 = sum(xor(S2_est_bit, x));
end

```

```

if m>2
S3_est_decimal = pskdemod(S3_est,M);
S3_est_bit = mary_2_bin(log2(M),S3_est_decimal);
number_of_bit_errors_3 = sum(xor(S3_est_bit, x));

```

```

end

if m>3
    S4_est_decimal = pskdemod(S4_est,M);
    S4_est_bit = mary_2_bin(log2(M),S4_est_decimal);
    number_of_bit_errors_4 = sum(xor(S4_est_bit, x));
end

% summing up the number of bit errors for the given SNR value
bit_errors_1 = bit_errors_1 + number_of_bit_errors_1;
bit_errors_2 = bit_errors_2 + number_of_bit_errors_2;
bit_errors_3 = bit_errors_3 + number_of_bit_errors_3;
bit_errors_4 = bit_errors_4 + number_of_bit_errors_4;

end % bit iteration ends

% saving the number of bit errors for the iterated SNR value
total_bit_errors_1 = [total_bit_errors_1 bit_errors_1];
total_bit_errors_2 = [total_bit_errors_2 bit_errors_2];
total_bit_errors_3 = [total_bit_errors_3 bit_errors_3];
total_bit_errors_4 = [total_bit_errors_4 bit_errors_4];

end % SNR iteration ends

% saving the number of bit errors for the iterated Monte Carlo run
total_bit_errors_monte_1 = [total_bit_errors_monte_1; total_bit_errors_1 ];
total_bit_errors_monte_2 = [total_bit_errors_monte_2; total_bit_errors_2 ];
total_bit_errors_monte_3 = [total_bit_errors_monte_3; total_bit_errors_3 ];
total_bit_errors_monte_4 = [total_bit_errors_monte_4; total_bit_errors_4 ];

end % Monte Carlo iteration ends

% computing the bit error rates
bit_error_rate_1 = total_bit_errors_monte_1 / n;
bit_error_rate_final_1 = mean(bit_error_rate_1);

```

```

bit_error_rate_2 = total_bit_errors_monte_2 / n;
bit_error_rate_final_2 = mean(bit_error_rate_2);

```

```

bit_error_rate_3 = total_bit_errors_monte_3 / n;
bit_error_rate_final_3 = mean(bit_error_rate_3);

```

```

bit_error_rate_4 = total_bit_errors_monte_4 / n;
bit_error_rate_final_4 = mean(bit_error_rate_4);

```

```

% plotting process, if only 1 Monte Carlo run is conducted, plot 'bit_error_rate_***'
% if more than 1 Monte Carlo run is conducted plot 'bit_error_rate_final_***'

```

```

if MonteCarlo == 1
    if m>0
        semilogy(EbNo_db,bit_error_rate_1,'oc-');grid on;hold on;legend('1x1'); end
    if m>1
        semilogy(EbNo_db,bit_error_rate_2,'xb-');grid on;hold on;legend('2x1','2x2'); end
    if m>2
        semilogy(EbNo_db,bit_error_rate_3,'dr-');grid on;hold on;legend('3x1','3x2', '3x3'); end
    if m>3
        semilogy(EbNo_db,bit_error_rate_4,'vg-');legend('4x1','4x2', '4x3','4x4'); end
else
    if m>0
        semilogy(EbNo_db,bit_error_rate_final_1,'oc-');grid on;hold on;legend('1x1'); end
    if m>1
        semilogy(EbNo_db,bit_error_rate_final_2,'xb-');grid on;hold on;legend('2x1','2x2'); end
    if m>2
        semilogy(EbNo_db,bit_error_rate_final_3,'dr-');grid on;hold on;legend('3x1','3x2', '3x3'); end
    if m>3
        semilogy(EbNo_db,bit_error_rate_final_4,'vg-');legend('4x1','4x2', '4x3','4x4'); end
end

```

```

title('MIMO system Performance', 'FontSize',12)

```

```

xlabel('E_b/N_o','FontSize',12);ylabel('BER','FontSize',12);

% legend('mx1','mx2', 'mx3','mx4');
%
%
% disp("'gtext" command is used here to add the simulation parameters on the graph');
% disp('waiting for you to click the mouse button');
% gtext(['Monte Carlo runs = ',num2str(MonteCarlo)]);
% gtext(['total bits sent for any SNR value ',num2str(n)]);
%
% gtext([num2str(M),'PSK']);
%

```

```

% Title      : This function simulates MIMO schemes:
%            MIMO system from mx1, mx2, until mxm (m <= 4)
%            Rate = 1/2 convolutional encoder and hard decision encoding is used.
%            BER curves of all schemes are presented in one figure
%            SUI Channel models are chosen as the simulation environment
% Author     : Nieh, Jo-Yen, Naval Postgraduate School, March 2006

%-----
% Simulation paramaters
clear all;clc;close all
m=4; % number of transmitting antenna
M=4; % m-ary PSK constellation, BPSK = 2
n=192*log2(M)*2*10; % total number of bits to be transmitted for the given SNR value
MonteCarlo = 1; % number of runs to be simulated

```

```

EbNo_db = 0:3:15; % SNR interval to be simulated
%-----
% Channel parameters
SUI_index = 2; % SUI model index, from 1 - 6
direct = 0; % antenna directivity, omni ==> 0, 30 deg directional ==> 1
% correlation parameters that can be used to adjust the degree of spatial correlation
rho_tx = 0.0; rho_rx = 0.0;
% sampling frequency, taken from the IEEE 802.16a OFDM channelization parameters for a
license-exempt band (BW = 20Mhz)
fs = 22.857; % Msps
%-----
% Noise parameters
sig = sqrt(0.5); % noise variance
No = 2*sig^2; % linear noise power
No_db = 10*log10(No); % noise power in dB
%-----
% Convolutional Encoder parameters
constlen = 7; % constraint length
codegen = [171 133]; % generator polynomials in octal form
trell = poly2trellis(constlen,codegen); % converting convolutional code polynomials to trellis
description.
%-----

awgn = 0; % to see AWGN performance awgn = 1;

% creating 'total bit errors array' for all Monte Carlo runs
total_bit_errors_monte_1 = []; total_bit_errors_monte_2 = [];
total_bit_errors_monte_3 = []; total_bit_errors_monte_4 = [];

for run = 1:MonteCarlo % Monte Carlo iteration

    % resetting 'total bit errors array' for all SNR values
    total_bit_errors_1 = []; total_bit_errors_2 = [];
    total_bit_errors_3 = []; total_bit_errors_4 = [];

```



```

for k=1:length(EbNo_db); % SNR iteration

    % resetting 'number of errors' for the given SNR value
    bit_errors_1 = 0; bit_errors_2 = 0; bit_errors_3 = 0; bit_errors_4 = 0;
    disp(['SNR = ',num2str(EbNo_db(k)), ' Monte Carlo run = ',num2str(run)]);

    for block=1:192*log2(M):n; % bit iteration

        % generating the transmitted symbols
        x = randint(1,192*log2(M));
        x_encoded = convenc(x,trel);
        x_decimal = bin_2_mary(log2(M),x_encoded); % binary to m-ary conversion 384
        x_mapped = pskmod(x_decimal,M);

        % STBC operation
        if m == 2
            [ y2 ] = STBC_encode_two(x_mapped);
            S1 = y2(1,1:192); S3 = y2(1,193:384);
            S2 = y2(2,1:192); S4 = y2(2,193:384);
        elseif m == 3
            [ y3 ] = STBC_encode_three(x_mapped);
            S1 = y3(1,1:128); S4 = y3(1,129:256); S7 = y3(1,257:384); S10 = y3(1,385:512);
            S2 = y3(2,1:128); S5 = y3(2,129:256); S8 = y3(2,257:384); S11 = y3(2,385:512);
            S3 = y3(3,1:128); S6 = y3(3,129:256); S9 = y3(3,257:384); S12 = y3(3,385:512);
        elseif m == 4
            [ y4 ] = STBC_encode_four(x_mapped);
            S1 = y4(1,1:96); S5 = y4(1,97:192); S9 = y4(1,193:288); S13 = y4(1,289:384); S17 = y4(1,385:480);
            S21 = y4(1,481:576); S25 = y4(1,577:672);
            S2 = y4(2,1:96); S6 = y4(2,97:192); S10 = y4(2,193:288); S14 = y4(2,289:384); S18 = y4(2,385:480);
            S22 = y4(2,481:576); S26 = y4(2,577:672);
            S3 = y4(3,1:96); S7 = y4(3,97:192); S11 = y4(3,193:288); S15 = y4(3,289:384); S19 = y4(3,385:480);
            S23 = y4(3,481:576); S27 = y4(3,577:672);
            S4 = y4(4,1:96); S8 = y4(4,97:192); S12 = y4(4,193:288); S16 = y4(4,289:384); S20 = y4(4,385:480);
            S24 = y4(4,481:576); S28 = y4(4,577:672);
        end
    end
end

```

```

% creating the channel
if awgn == 0;
    [H1 H2 H3 H4] = build_mimo_channel (SUI_index,direct,fs,m,rho_tx,rho_rx);
else
    H1 = {[1]};
    H2 = {[1]};
    H3 = {[1]};
    H4 = {[1]};
end

% Es(linear) computation with respect to the noise power
EsNo_db = EbNo_db(k) + 10*log10(log2(M));
Es_db = EsNo_db + No_db;
Es = 10.^(Es_db/10);

if m == 1
    % 1x1 received signals
    R1{1,1} = add_noise2sequ(sqrt(Es)*filter(H1{1},1,[S1 S2]),1,sig);

elseif m == 2
    [a b]=size(S1);
    R3=[zeros(a,b) [zeros(a,b)] ;
        [zeros(a,b) [zeros(a,b)] ];

    % 2x1 received signals
    R2{1,1} = add_noise2sequ(sqrt(Es/2)*filter(H2{1},1,S1) + sqrt(Es/2)*filter(H2{2},1,S2),1,sig);
    R2{1,2} = add_noise2sequ(sqrt(Es/2)*filter(H2{1},1,S3) + sqrt(Es/2)*filter(H2{2},1,S4),1,sig);

    % 2x2 received signals
    for i = 1:2
        R2{2,2*(i-1)+1} = add_noise2sequ(sqrt(Es/2)*filter(H2{2*(i-1)+1},1,S1) +
sqrt(Es/2)*filter(H2{2*(i-1)+2},1,S2),1,sig);
        R2{2,2*(i-1)+2} = add_noise2sequ(sqrt(Es/2)*filter(H2{2*(i-1)+1},1,S3) +
sqrt(Es/2)*filter(H2{2*(i-1)+2},1,S4),1,sig);
    end

```

```

end

elseif m == 3
    % 3x1 received signals
    R3{1,1} = add_noise2sequ( sqrt(Es/3)*filter(H1{1},1,S1 ) + sqrt(Es/3)*filter(H1{2},1,S2 ) +
    sqrt(Es/3)*filter(H1{3},1,S3 ),1,sig);
    R3{1,2} = add_noise2sequ( sqrt(Es/3)*filter(H1{1},1,S4 ) + sqrt(Es/3)*filter(H1{2},1,S5 ) +
    sqrt(Es/3)*filter(H1{3},1,S6 ),1,sig);
    R3{1,3} = add_noise2sequ( sqrt(Es/3)*filter(H1{1},1,S7 ) + sqrt(Es/3)*filter(H1{2},1,S8 ) +
    sqrt(Es/3)*filter(H1{3},1,S9 ),1,sig);
    R3{1,4} = add_noise2sequ( sqrt(Es/3)*filter(H1{1},1,S10) + sqrt(Es/3)*filter(H1{2},1,S11) +
    sqrt(Es/3)*filter(H1{3},1,S12),1,sig);

    % 3x2 received signals
    for i = 1:2
        R3{2,4*(i-1)+1} = add_noise2sequ( sqrt(Es/3)*filter(H2{3*(i-1)+1},1,S1 ) +
        sqrt(Es/3)*filter(H2{3*(i-1)+2},1,S2 ) + sqrt(Es/3)*filter(H2{3*(i-1)+3},1,S3 ),1,sig);
        R3{2,4*(i-1)+2} = add_noise2sequ( sqrt(Es/3)*filter(H2{3*(i-1)+1},1,S4 ) +
        sqrt(Es/3)*filter(H2{3*(i-1)+2},1,S5 ) + sqrt(Es/3)*filter(H2{3*(i-1)+3},1,S6 ),1,sig);
        R3{2,4*(i-1)+3} = add_noise2sequ( sqrt(Es/3)*filter(H2{3*(i-1)+1},1,S7 ) +
        sqrt(Es/3)*filter(H2{3*(i-1)+2},1,S8 ) + sqrt(Es/3)*filter(H2{3*(i-1)+3},1,S9 ),1,sig);
        R3{2,4*(i-1)+4} = add_noise2sequ( sqrt(Es/3)*filter(H2{3*(i-1)+1},1,S10) +
        sqrt(Es/3)*filter(H2{3*(i-1)+2},1,S11) + sqrt(Es/3)*filter(H2{3*(i-1)+3},1,S12),1,sig);
    end

    % 3x3 received signals
    for i = 1:3
        R3{3,4*(i-1)+1} = add_noise2sequ( sqrt(Es/3)*filter(H3{3*(i-1)+1},1,S1 ) +
        sqrt(Es/3)*filter(H3{3*(i-1)+2},1,S2 ) + sqrt(Es/3)*filter(H3{3*(i-1)+3},1,S3 ),1,sig);
        R3{3,4*(i-1)+2} = add_noise2sequ( sqrt(Es/3)*filter(H3{3*(i-1)+1},1,S4 ) +
        sqrt(Es/3)*filter(H3{3*(i-1)+2},1,S5 ) + sqrt(Es/3)*filter(H3{3*(i-1)+3},1,S6 ),1,sig);
        R3{3,4*(i-1)+3} = add_noise2sequ( sqrt(Es/3)*filter(H3{3*(i-1)+1},1,S7 ) +
        sqrt(Es/3)*filter(H3{3*(i-1)+2},1,S8 ) + sqrt(Es/3)*filter(H3{3*(i-1)+3},1,S9 ),1,sig);
        R3{3,4*(i-1)+4} = add_noise2sequ( sqrt(Es/3)*filter(H3{3*(i-1)+1},1,S10) +
        sqrt(Es/3)*filter(H3{3*(i-1)+2},1,S11) + sqrt(Es/3)*filter(H3{3*(i-1)+3},1,S12),1,sig);
    end
end

```

```

elseif m == 4
% 4x1 received signals
R4{1,1} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S1 ) + sqrt(Es/4)*filter(H1{2},1,S2 ) +
sqrt(Es/4)*filter(H1{3},1,S3 ) + sqrt(Es/4)*filter(H1{4},1,S4 ),1,sig);
R4{1,2} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S5 ) + sqrt(Es/4)*filter(H1{2},1,S6 ) +
sqrt(Es/4)*filter(H1{3},1,S7 ) + sqrt(Es/4)*filter(H1{4},1,S8 ),1,sig);
R4{1,3} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S9 ) + sqrt(Es/4)*filter(H1{2},1,S10) +
sqrt(Es/4)*filter(H1{3},1,S11) + sqrt(Es/4)*filter(H1{4},1,S12),1,sig);
R4{1,4} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S13) + sqrt(Es/4)*filter(H1{2},1,S14) +
sqrt(Es/4)*filter(H1{3},1,S15) + sqrt(Es/4)*filter(H1{4},1,S16),1,sig);
R4{1,5} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S17) + sqrt(Es/4)*filter(H1{2},1,S18) +
sqrt(Es/4)*filter(H1{3},1,S19) + sqrt(Es/4)*filter(H1{4},1,S20),1,sig);
R4{1,6} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S21) + sqrt(Es/4)*filter(H1{2},1,S22) +
sqrt(Es/4)*filter(H1{3},1,S23) + sqrt(Es/4)*filter(H1{4},1,S24),1,sig);
R4{1,7} = add_noise2sequ( sqrt(Es/4)*filter(H1{1},1,S25) + sqrt(Es/4)*filter(H1{2},1,S26) +
sqrt(Es/4)*filter(H1{3},1,S27) + sqrt(Es/4)*filter(H1{4},1,S28),1,sig);

% 4x2 received signals
for i = 1:2
R4{2,7*(i-1)+1} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S1 ) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S2 ) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S3 ) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S4 ),1,sig);
R4{2,7*(i-1)+2} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S5 ) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S6 ) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S7 ) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S8 ),1,sig);
R4{2,7*(i-1)+3} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S9 ) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S10) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S11) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S12),1,sig);
R4{2,7*(i-1)+4} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S13) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S14) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S15) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S16),1,sig);
R4{2,7*(i-1)+5} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S17) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S18) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S19) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S20),1,sig);
R4{2,7*(i-1)+6} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S21) +

```

```

sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S22) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S23) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S24),1,sig);
    R4{2,7*(i-1)+7} = add_noise2sequ( sqrt(Es/4)*filter(H2{4*(i-1)+1},1,S25) +
sqrt(Es/4)*filter(H2{4*(i-1)+2},1,S26) + sqrt(Es/4)*filter(H2{4*(i-1)+3},1,S27) +
sqrt(Es/4)*filter(H2{4*(i-1)+4},1,S28),1,sig);
end

```

**% 4x3 received signals**

**for i = 1:3**

```

    R4{3,7*(i-1)+1} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S1 ) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S2 ) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S3 ) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S4 ),1,sig);
    R4{3,7*(i-1)+2} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S5 ) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S6 ) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S7 ) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S8 ),1,sig);
    R4{3,7*(i-1)+3} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S9 ) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S10) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S11) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S12),1,sig);
    R4{3,7*(i-1)+4} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S13) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S14) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S15) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S16),1,sig);
    R4{3,7*(i-1)+5} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S17) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S18) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S19) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S20),1,sig);
    R4{3,7*(i-1)+6} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S21) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S22) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S23) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S24),1,sig);
    R4{3,7*(i-1)+7} = add_noise2sequ( sqrt(Es/4)*filter(H3{4*(i-1)+1},1,S25) +
sqrt(Es/4)*filter(H3{4*(i-1)+2},1,S26) + sqrt(Es/4)*filter(H3{4*(i-1)+3},1,S27) +
sqrt(Es/4)*filter(H3{4*(i-1)+4},1,S28),1,sig);
end

```

**% 4x4 received signals**

**for i = 1:4**

```

    R4{4,7*(i-1)+1} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S1 ) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S2 ) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S3 ) +

```

```

sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S4 ),1,sig);
    R4{4,7*(i-1)+2} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S5 ) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S6 ) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S7 ) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S8 ),1,sig);
    R4{4,7*(i-1)+3} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S9 ) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S10) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S11) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S12),1,sig);
    R4{4,7*(i-1)+4} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S13) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S14) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S15) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S16),1,sig);
    R4{4,7*(i-1)+5} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S17) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S18) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S19) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S20),1,sig);
    R4{4,7*(i-1)+6} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S21) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S22) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S23) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S24),1,sig);
    R4{4,7*(i-1)+7} = add_noise2sequ( sqrt(Es/4)*filter(H4{4*(i-1)+1},1,S25) +
sqrt(Es/4)*filter(H4{4*(i-1)+2},1,S26) + sqrt(Es/4)*filter(H4{4*(i-1)+3},1,S27) +
sqrt(Es/4)*filter(H4{4*(i-1)+4},1,S28),1,sig);
end

end

% computing the estimates
if m ==1
    S1_est{1,1} = non_causal_filter(conj(H1{1}),R1{1,1});

elseif m == 2
    % 2x1 estimated signals
    [a,b]=size(R2{1,1});
    S2_est=[zeros(a,b)] [zeros(a,b)] ;[zeros(a,b)] [zeros(a,b)];
    S2_est{1,1} = non_causal_filter(conj(H1{1}),R2{1,1}) + filter( H1{2},1,conj(R2{1,2}));
    S2_est{1,2} = non_causal_filter(conj(H1{2}),R2{1,1}) + filter(-H1{1},1,conj(R2{1,2}));

    % 2x2 estimated signals
    for i = 1 : 2

```

```

S2_est{2,1} = S2_est{2,1} + non_causal_filter(conj(H2{2*(i-1)+1}),R2{2,2*(i-1)+1}) +
filter( H2{2*(i-1)+2},1,conj(R2{2,2*(i-1)+2}));

S2_est{2,2} = S2_est{2,2} + non_causal_filter(conj(H2{2*(i-1)+2}),R2{2,2*(i-1)+1}) + filter(-
H2{2*(i-1)+1},1,conj(R2{2,2*(i-1)+2}));

end

elseif m == 3

% 3x1 estimated signals
[a,b]=size(R3{1,1});
S3_est=[zeros(a,b) [zeros(a,b)] [zeros(a,b)] ;[zeros(a,b)] [zeros(a,b)]
[zeros(a,b)];[zeros(a,b)] [zeros(a,b)] [zeros(a,b)];];
S3_est{1,1} = non_causal_filter(conj(H1{1}),R3{1,1}) + filter( H1{2},1,conj(R3{1,2})) +
filter( H1{3},1,conj(R3{1,3}));
S3_est{1,2} = non_causal_filter(conj(H1{2}),R3{1,1}) + filter(-H1{1},1,conj(R3{1,2})) +
filter( H1{3},1,conj(R3{1,4}));
S3_est{1,3} = non_causal_filter(conj(H1{3}),R3{1,1}) + filter(-H1{1},1,conj(R3{1,3})) + filter(-
H1{2},1,conj(R3{1,4}));

% 3x2 estimated signals
for i = 1 : 2
S3_est{2,1} = S3_est{2,1} + non_causal_filter(conj(H2{3*(i-1)+1}),R3{2,4*(i-1)+1}) +
filter( H2{3*(i-1)+2},1,conj(R3{2,4*(i-1)+2})) + filter( H2{3*(i-1)+3},1,conj(R3{2,4*(i-1)+3}));
S3_est{2,2} = S3_est{2,2} + non_causal_filter(conj(H2{3*(i-1)+2}),R3{2,4*(i-1)+1}) + filter(-
H2{3*(i-1)+1},1,conj(R3{2,4*(i-1)+2})) + filter( H2{3*(i-1)+3},1,conj(R3{2,4*(i-1)+4}));
S3_est{2,3} = S3_est{2,3} + non_causal_filter(conj(H2{3*(i-1)+3}),R3{2,4*(i-1)+1}) + filter(-
H2{3*(i-1)+1},1,conj(R3{2,4*(i-1)+3})) + filter(-H2{3*(i-1)+2},1,conj(R3{2,4*(i-1)+4}));
end

% 3x3 estimated signals
for i = 1 : 3
S3_est{3,1} = S3_est{3,1} + non_causal_filter(conj(H3{3*(i-1)+1}),R3{3,4*(i-1)+1}) +
filter( H3{3*(i-1)+2},1,conj(R3{3,4*(i-1)+2})) + filter( H3{3*(i-1)+3},1,conj(R3{3,4*(i-1)+3}));
S3_est{3,2} = S3_est{3,2} + non_causal_filter(conj(H3{3*(i-1)+2}),R3{3,4*(i-1)+1}) + filter(-
H3{3*(i-1)+1},1,conj(R3{3,4*(i-1)+2})) + filter( H3{3*(i-1)+3},1,conj(R3{3,4*(i-1)+4}));
S3_est{3,3} = S3_est{3,3} + non_causal_filter(conj(H3{3*(i-1)+3}),R3{3,4*(i-1)+1}) + filter(-
H3{3*(i-1)+1},1,conj(R3{3,4*(i-1)+3})) + filter(-H3{3*(i-1)+2},1,conj(R3{3,4*(i-1)+4}));

```

```

end

elseif m == 4
% 4x1 estimated signals
[a,b]=size(R4{1,1});
S4_est=[zeros(a,b)] [zeros(a,b)] [zeros(a,b)] [zeros(a,b)];[zeros(a,b)] [zeros(a,b)] [zeros(a,b)]
[zeros(a,b)];
[zeros(a,b)] [zeros(a,b)] [zeros(a,b)] [zeros(a,b)];[zeros(a,b)] [zeros(a,b)] [zeros(a,b)]
[zeros(a,b)];
S4_est{1,1} = non_causal_filter(conj(H1{1}),R4{1,1}) + filter( H1{2},1,conj(R4{1,2})) +
filter( H1{3},1,conj(R4{1,3})) + filter( H1{4},1,conj(R4{1,4}));
S4_est{1,2} = non_causal_filter(conj(H1{2}),R4{1,1}) + filter(-H1{1},1,conj(R4{1,2})) +
filter( H1{3},1,conj(R4{1,5})) + filter( H1{4},1,conj(R4{1,6}));
S4_est{1,3} = non_causal_filter(conj(H1{3}),R4{1,1}) + filter(-H1{1},1,conj(R4{1,3})) + filter(-
H1{2},1,conj(R4{1,5})) + filter( H1{4},1,conj(R4{1,7}));
S4_est{1,4} = non_causal_filter(conj(H1{4}),R4{1,1}) + filter(-H1{1},1,conj(R4{1,4})) + filter(-
H1{2},1,conj(R4{1,6})) + filter(-H1{3},1,conj(R4{1,7}));

% 4x2 estimated signals
for i = 1 : 2
S4_est{2,1} = S4_est{2,1} + non_causal_filter(conj(H2{4*(i-1)+1}),R4{2,7*(i-1)+1}) +
filter( H2{4*(i-1)+2},1,conj(R4{2,7*(i-1)+2})) + filter( H2{4*(i-1)+3},1,conj(R4{2,7*(i-1)+3})) +
filter( H2{4*(i-1)+4},1,conj(R4{2,7*(i-1)+4}));
S4_est{2,2} = S4_est{2,2} + non_causal_filter(conj(H2{4*(i-1)+2}),R4{2,7*(i-1)+1}) + filter(-
H2{4*(i-1)+1},1,conj(R4{2,7*(i-1)+2})) + filter( H2{4*(i-1)+3},1,conj(R4{2,7*(i-1)+5})) +
filter( H2{4*(i-1)+4},1,conj(R4{2,7*(i-1)+6}));
S4_est{2,3} = S4_est{2,3} + non_causal_filter(conj(H2{4*(i-1)+3}),R4{2,7*(i-1)+1}) + filter(-
H2{4*(i-1)+1},1,conj(R4{2,7*(i-1)+3})) + filter(-H2{4*(i-1)+2},1,conj(R4{2,7*(i-1)+5})) +
filter( H2{4*(i-1)+4},1,conj(R4{2,7*(i-1)+7}));
S4_est{2,4} = S4_est{2,4} + non_causal_filter(conj(H2{4*(i-1)+4}),R4{2,7*(i-1)+1}) + filter(-
H2{4*(i-1)+1},1,conj(R4{2,7*(i-1)+4})) + filter(-H2{4*(i-1)+2},1,conj(R4{2,7*(i-1)+6})) + filter(-
H2{4*(i-1)+3},1,conj(R4{2,7*(i-1)+7}));
end

% 4x3 estimated signals
for i = 1 : 3

```



```

S4_est{3,1} = S4_est{3,1} + non_causal_filter(conj(H3{4*(i-1)+1}),R4{3,7*(i-1)+1}) +
filter( H3{4*(i-1)+2},1,conj(R4{3,7*(i-1)+2})) + filter( H3{4*(i-1)+3},1,conj(R4{3,7*(i-1)+3})) +
filter( H3{4*(i-1)+4},1,conj(R4{3,7*(i-1)+4}));

S4_est{3,2} = S4_est{3,2} + non_causal_filter(conj(H3{4*(i-1)+2}),R4{3,7*(i-1)+1}) + filter(-
H3{4*(i-1)+1},1,conj(R4{3,7*(i-1)+2})) + filter( H3{4*(i-1)+3},1,conj(R4{3,7*(i-1)+5})) +
filter( H3{4*(i-1)+4},1,conj(R4{3,7*(i-1)+6}));

S4_est{3,3} = S4_est{3,3} + non_causal_filter(conj(H3{4*(i-1)+3}),R4{3,7*(i-1)+1}) + filter(-
H3{4*(i-1)+1},1,conj(R4{3,7*(i-1)+3})) + filter(-H3{4*(i-1)+2},1,conj(R4{3,7*(i-1)+5})) +
filter( H3{4*(i-1)+4},1,conj(R4{3,7*(i-1)+7}));

S4_est{3,4} = S4_est{3,4} + non_causal_filter(conj(H3{4*(i-1)+4}),R4{3,7*(i-1)+1}) + filter(-
H3{4*(i-1)+1},1,conj(R4{3,7*(i-1)+4})) + filter(-H3{4*(i-1)+2},1,conj(R4{3,7*(i-1)+6})) + filter(-
H3{4*(i-1)+3},1,conj(R4{3,7*(i-1)+7}));

end

```

**% 4x4 estimated signals**

```

for i = 1 : 4

S4_est{4,1} = S4_est{4,1} + non_causal_filter(conj(H4{4*(i-1)+1}),R4{4,7*(i-1)+1}) +
filter( H4{4*(i-1)+2},1,conj(R4{4,7*(i-1)+2})) + filter( H4{4*(i-1)+3},1,conj(R4{4,7*(i-1)+3})) +
filter( H4{4*(i-1)+4},1,conj(R4{4,7*(i-1)+4}));

S4_est{4,2} = S4_est{4,2} + non_causal_filter(conj(H4{4*(i-1)+2}),R4{4,7*(i-1)+1}) + filter(-
H4{4*(i-1)+1},1,conj(R4{4,7*(i-1)+2})) + filter( H4{4*(i-1)+3},1,conj(R4{4,7*(i-1)+5})) +
filter( H4{4*(i-1)+4},1,conj(R4{4,7*(i-1)+6}));

S4_est{4,3} = S4_est{4,3} + non_causal_filter(conj(H4{4*(i-1)+3}),R4{4,7*(i-1)+1}) + filter(-
H4{4*(i-1)+1},1,conj(R4{4,7*(i-1)+3})) + filter(-H4{4*(i-1)+2},1,conj(R4{4,7*(i-1)+5})) +
filter( H4{4*(i-1)+4},1,conj(R4{4,7*(i-1)+7}));

S4_est{4,4} = S4_est{4,4} + non_causal_filter(conj(H4{4*(i-1)+4}),R4{4,7*(i-1)+1}) + filter(-
H4{4*(i-1)+1},1,conj(R4{4,7*(i-1)+4})) + filter(-H4{4*(i-1)+2},1,conj(R4{4,7*(i-1)+6})) + filter(-
H4{4*(i-1)+3},1,conj(R4{4,7*(i-1)+7}));

end

end

```

**% Recollecting the estimates in a single sequence for demapping**

```

if m == 1
S_est = [S1_est{1,1}];
elseif m == 2

```

```

S_est = [S2_est{1,1} S2_est{1,2};
         S2_est{2,1} S2_est{2,2}];
elseif m == 3
S_est = [S3_est{1,1} S3_est{1,2} S3_est{1,3};
         S3_est{2,1} S3_est{2,2} S3_est{2,3};
         S3_est{3,1} S3_est{3,2} S3_est{3,3}];
elseif m == 4
S_est = [S4_est{1,1} S4_est{1,2} S4_est{1,3} S4_est{1,4};
         S4_est{2,1} S4_est{2,2} S4_est{2,3} S4_est{2,4};
         S4_est{3,1} S4_est{3,2} S4_est{3,3} S4_est{3,4};
         S4_est{4,1} S4_est{4,2} S4_est{4,3} S4_est{4,4}];
end

% demapping, decoding and bit comparison for single carrier systems
if m > 0
r_decimal_1 = pskdemod(S_est(1,:),M);
r_bits_1 = mary_2_bin(log2(M),r_decimal_1);
r_decoded_1 = vitdec(r_bits_1, trel, 1, 'trunc', 'hard');
number_of_bit_errors_1 = sum(xor(r_decoded_1, x));
number_of_bit_errors_2=0;number_of_bit_errors_3=0;number_of_bit_errors_4=0;
end
if m > 1
r_decimal_2 = pskdemod(S_est(2,:),M);
r_bits_2 = mary_2_bin(log2(M),r_decimal_2);
r_decoded_2 = vitdec(r_bits_2, trel, 1, 'trunc', 'hard');
number_of_bit_errors_2 = sum(xor(r_decoded_2, x));
number_of_bit_errors_3=0;number_of_bit_errors_4=0;
end
if m > 2
r_decimal_3 = pskdemod(S_est(3,:),M);
r_bits_3 = mary_2_bin(log2(M),r_decimal_3);
r_decoded_3 = vitdec(r_bits_3, trel, 1, 'trunc', 'hard');
number_of_bit_errors_3 = sum(xor(r_decoded_3, x));
number_of_bit_errors_4=0;
end
if m > 3

```

```

r_decimal_4 = pskdemod(S_est(4,:),M);
r_bits_4 = mary_2_bin(log2(M),r_decimal_4);
r_decoded_4 = vitdec(r_bits_4, trel, 1, 'trunc', 'hard');
number_of_bit_errors_4 = sum(xor(r_decoded_4, x));
end

% summing up the number of bit errors for the given SNR value
bit_errors_1 = bit_errors_1 + number_of_bit_errors_1;
bit_errors_2 = bit_errors_2 + number_of_bit_errors_2;
bit_errors_3 = bit_errors_3 + number_of_bit_errors_3;
bit_errors_4 = bit_errors_4 + number_of_bit_errors_4;

end % bit iteration ends

% saving the number of bit errors for the iterated SNR value
total_bit_errors_1 = [total_bit_errors_1 bit_errors_1];
total_bit_errors_2 = [total_bit_errors_2 bit_errors_2];
total_bit_errors_3 = [total_bit_errors_3 bit_errors_3];
total_bit_errors_4 = [total_bit_errors_4 bit_errors_4];

end % SNR iteration ends

% saving the number of bit errors for the iterated Monte Carlo run
total_bit_errors_monte_1 = [total_bit_errors_monte_1; total_bit_errors_1 ];
total_bit_errors_monte_2 = [total_bit_errors_monte_2; total_bit_errors_2 ];
total_bit_errors_monte_3 = [total_bit_errors_monte_3; total_bit_errors_3 ];
total_bit_errors_monte_4 = [total_bit_errors_monte_4; total_bit_errors_4 ];

end % Monte Carlo iteration ends

% computing the bit error rates
bit_error_rate_1 = total_bit_errors_monte_1 / n;
bit_error_rate_final_1 = mean(bit_error_rate_1);

bit_error_rate_2 = total_bit_errors_monte_2 / n;

```

```
bit_error_rate_final_2 = mean(bit_error_rate_2);
```

```
bit_error_rate_3 = total_bit_errors_monte_3 / n;
```

```
bit_error_rate_final_3 = mean(bit_error_rate_3);
```

```
bit_error_rate_4 = total_bit_errors_monte_4 / n;
```

```
bit_error_rate_final_4 = mean(bit_error_rate_4);
```

```
if MonteCarlo == 1
```

```
    figure
```

```
    if m>0
```

```
        semilogy(EbNo_db,bit_error_rate_1,'^k--');grid on;hold on;
```

```
        legend('1x1')
```

```
    end
```

```
    if m>1
```

```
        semilogy(EbNo_db,bit_error_rate_2,'ob--');grid on;hold on;
```

```
        legend('2x1','2x2')
```

```
    end
```

```
    if m>2
```

```
        semilogy(EbNo_db,bit_error_rate_3,'dr--');grid on;hold on;
```

```
        legend('3x1','3x2','3x3')
```

```
    end
```

```
    if m>3
```

```
        semilogy(EbNo_db,bit_error_rate_4,'sg--');
```

```
        legend('4x1','4x2','4x3','4x4')
```

```
    end
```

```
else
```

```
    figure
```

```
    if m>0
```

```
        semilogy(EbNo_db,bit_error_rate_final_1,'^k--');grid on; hold on;
```

```
        legend('1x1')
```

```
    end
```

```
    if m>1
```

```
        semilogy(EbNo_db,bit_error_rate_final_2,'ob--');grid on; hold on;
```

```

        legend('2x1','2x2')
    end
    if m>2
        semilogy(EbNo_db,bit_error_rate_final_3,'dr--');grid on; hold on;
        legend('3x1','3x2','3x3')
    end
    if m>3
        semilogy(EbNo_db,bit_error_rate_final_4,'sg--');
        legend('4x1','4x2','4x3','4x4')
    end

end

title('MIMO scheme with STBC over SUI index = 2 and rho_tx =0.0')
xlabel('E_b/N_o');ylabel('BER');

```

```

% Title      : This function compares channel coorelation with MISO schemes
%            : Correlation coefficients comparisons with MISO scheme
%            : Rate = 1/2 convolutional encoder and hard decision encoding is used.
%            : BER curves of all schemes are presented in one figure
%            : SUI Channel models are chosen as the simulation environment
% Author     : Nieh, Jo-Yen, Naval Postgraduate School, March 2006

%-----

```

```

% Simulation paramaters
clear all;clc;close all
m=4; % number of transmit antenna
M=4; % m-ary PSK constellation, BPSK = 2
n=192*log2(M)*2*10000*2; % total number of bits to be transmitted for the given SNR value
MonteCarlo = 12; % number of runs to be simulated
EbNo_db =0:2:16; % SNR interval to be simulated
%-----
% Channel parameters
SUI_index = 2; % SUI model index, from 1 - 6
direct = 1; % antenna directivity, omni ==> 0, 30 deg directional ==> 1
% correlation parameters that can be used to adjust the degree of spatial correlation
rho_tx=[0 0.1 0.2 0.4 0.6 0.8 1]; % eight different comparisons
% sampling frequency, taken from the IEEE 802.16a OFDM channelization parameters for a
license-exempt band (BW = 20Mhz)
fs = 22.857; % Msps
%-----
% Noise parameters
sig = sqrt(0.5); % noise variance
No = 2*sig^2; % linear noise power
No_db = 10*log10(No); % noise power in dB
%-----
% Convolutional Encoder parameters
constlen =7; % constraint length
codegen = [171 133]; % generator polynomials in octal form
trel = poly2trellis(constlen,codegen); % converting convolutional code polynomials to trellis
description.
%-----

awgn = 0; % to see AWGN performance awgn = 1;

% creating 'total bit errors array' for all Monte Carlo runs
total_bit_errors_monte_1 = []; total_bit_errors_monte_2 = [];
total_bit_errors_monte_3 = []; total_bit_errors_monte_4 = [];
total_bit_errors_monte_5 = []; total_bit_errors_monte_6 = [];
total_bit_errors_monte_7 = []; total_bit_errors_monte_8 = [];

```

```

for run = 1:MonteCarlo % Monte Carlo iteration

% resetting 'total bit errors array' for all SNR values
total_bit_errors_1 = []; total_bit_errors_2 = [];
total_bit_errors_3 = []; total_bit_errors_4 = [];
total_bit_errors_5 = []; total_bit_errors_6 = [];
total_bit_errors_7 = []; total_bit_errors_8 = [];

for k=1:length(EbNo_db); % SNR iteration

% resetting 'number of errors' for the given SNR value
bit_errors_1 = 0; bit_errors_2 = 0; bit_errors_3 = 0; bit_errors_4 = 0;
bit_errors_5 = 0; bit_errors_6 = 0; bit_errors_7 = 0; bit_errors_8 = 0;
disp(['SNR = ',num2str(EbNo_db(k)), ' Monte Carlo run = ',num2str(run)]);

for block=1:192*log2(M):n; % bit iteration

% generating the transmitted symbols
x = randint(1,192*log2(M));
x_encoded = convenc(x,trel);
x_decimal = bin_2_mary(log2(M),x_encoded); % binary to m-ary conversion 384
x_mapped = pskmod(x_decimal,M);

% STBC operation
if m == 2
[ y2 ] = STBC_encode_two(x_mapped);
S1 = y2(1,1:192); S3 = y2(1,193:384);
S2 = y2(2,1:192); S4 = y2(2,193:384);
elseif m == 3
[ y3 ] = STBC_encode_three(x_mapped);
T1 = y3(1,1:128); T4 = y3(1,129:256); T7 = y3(1,257:384); T10 = y3(1,385:512);
T2 = y3(2,1:128); T5 = y3(2,129:256); T8 = y3(2,257:384); T11 = y3(2,385:512);
T3 = y3(3,1:128); T6 = y3(3,129:256); T9 = y3(3,257:384); T12 = y3(3,385:512);
elseif m == 4

```

```

[ y4 ] = STBC_encode_four(x_mapped);
U1 = y4(1,1:96); U5 = y4(1,97:192); U9 = y4(1,193:288); U13 = y4(1,289:384); U17 =
y4(1,385:480); U21 = y4(1,481:576); U25 = y4(1,577:672);
U2 = y4(2,1:96); U6 = y4(2,97:192); U10 = y4(2,193:288); U14 = y4(2,289:384); U18 =
y4(2,385:480); U22 = y4(2,481:576); U26 = y4(2,577:672);
U3 = y4(3,1:96); U7 = y4(3,97:192); U11 = y4(3,193:288); U15 = y4(3,289:384); U19 =
y4(3,385:480); U23 = y4(3,481:576); U27 = y4(3,577:672);
U4 = y4(4,1:96); U8 = y4(4,97:192); U12 = y4(4,193:288); U16 = y4(4,289:384); U20 =
y4(4,385:480); U24 = y4(4,481:576); U28 = y4(4,577:672);
end

% creating the channel
if awgn == 0;
    if m == 1
        for i=1:length(rho_tx)
            [h1x1_1] = build_MISO_channel (SUI_index,direct,fs,m,rho_tx(i));
            H(i,:)= {h1x1_1};
        end
    elseif m == 2
        for i=1:length(rho_tx)
            [h2x1_1,h2x1_2] = build_MISO_channel (SUI_index,direct,fs,m,rho_tx(i));
            H(i,:)= {h2x1_1 h2x1_2};
        end
    elseif m == 3
        for i=1:length(rho_tx)
            [h3x1_1,h3x1_2,h3x1_3] = build_MISO_channel (SUI_index,direct,fs,m,rho_tx(i));
            H(i,:)= {h3x1_1 h3x1_2 h3x1_3};
        end
    elseif m == 4
        for i=1:length(rho_tx)
            [h4x1_1,h4x1_2,h4x1_3,h4x1_4] = build_MISO_channel
(SUI_index,direct,fs,m,rho_tx(i));
            H(i,:)= {h4x1_1 h4x1_2 h4x1_3 h4x1_4};
        end
    end
end

```



```

end

else
    if m == 1
        for i=1:length(rho_tx)
            H(i,:)= {[1]};
        end
    elseif m == 2
        for i=1:length(rho_tx)
            H(i,:)= {[1] [1]};
        end
    elseif m == 3
        for i=1:length(rho_tx)
            H(i,:)= {[1] [1] [1]};
        end
    elseif m == 4
        for i=1:length(rho_tx)
            H(i,:)= {[1] [1] [1] [1]};
        end
    end
end

% Es(linear) computation with respect to the noise power
EsNo_db = EbNo_db(k) + 10*log10(log2(M));
Es_db = EsNo_db + No_db;
Es = 10.^(Es_db/10);

if m == 1
    % 1x1 received signals
    R1{i} = add_noise2sequ(sqrt(Es)*filter(H{i,1},1,[S1 S2]),1,sig);

elseif m == 2
    % 2x1 received signals
    for i=1:length(rho_tx)
        R1{i} = add_noise2sequ(sqrt(Es/2)*filter(H{i,1},1,S1) + sqrt(Es/2)*filter(H{i,2},1,S2),1,sig);
        R2{i} = add_noise2sequ(sqrt(Es/2)*filter(H{i,1},1,S3) + sqrt(Es/2)*filter(H{i,2},1,S4),1,sig);
    end
end

```

```

end

elseif m == 3
    % 3x1 received signals
    for i=1:length(rho_tx)
        R1{i} = add_noise2sequ( sqrt(Es/3)*filter(H{i,1},1,T1 ) + sqrt(Es/3)*filter(H{i,2},1,T2 ) +
sqrt(Es/3)*filter(H{i,3},1,T3 ),1,sig);
        R2{i} = add_noise2sequ( sqrt(Es/3)*filter(H{i,1},1,T4 ) + sqrt(Es/3)*filter(H{i,2},1,T5 ) +
sqrt(Es/3)*filter(H{i,3},1,T6 ),1,sig);
        R3{i} = add_noise2sequ( sqrt(Es/3)*filter(H{i,1},1,T7 ) + sqrt(Es/3)*filter(H{i,2},1,T8 ) +
sqrt(Es/3)*filter(H{i,3},1,T9 ),1,sig);
        R4{i} = add_noise2sequ( sqrt(Es/3)*filter(H{i,1},1,T10) + sqrt(Es/3)*filter(H{i,2},1,T11) +
sqrt(Es/3)*filter(H{i,3},1,T12),1,sig);
    end

elseif m == 4
    % 4x1 received signals
    for i=1:length(rho_tx)
        R1{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U1 ) + sqrt(Es/4)*filter(H{i,2},1,U2 ) +
sqrt(Es/4)*filter(H{i,3},1,U3 ) + sqrt(Es/4)*filter(H{i,4},1,U4 ),1,sig);
        R2{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U5 ) + sqrt(Es/4)*filter(H{i,2},1,U6 ) +
sqrt(Es/4)*filter(H{i,3},1,U7 ) + sqrt(Es/4)*filter(H{i,4},1,U8 ),1,sig);
        R3{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U9 ) + sqrt(Es/4)*filter(H{i,2},1,U10) +
sqrt(Es/4)*filter(H{i,3},1,U11) + sqrt(Es/4)*filter(H{i,4},1,U12),1,sig);
        R4{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U13) + sqrt(Es/4)*filter(H{i,2},1,U14) +
sqrt(Es/4)*filter(H{i,3},1,U15) + sqrt(Es/4)*filter(H{i,4},1,U16),1,sig);
        R5{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U17) + sqrt(Es/4)*filter(H{i,2},1,U18) +
sqrt(Es/4)*filter(H{i,3},1,U19) + sqrt(Es/4)*filter(H{i,4},1,U20),1,sig);
        R6{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U21) + sqrt(Es/4)*filter(H{i,2},1,U22) +
sqrt(Es/4)*filter(H{i,3},1,U23) + sqrt(Es/4)*filter(H{i,4},1,U24),1,sig);
        R7{i} = add_noise2sequ( sqrt(Es/4)*filter(H{i,1},1,U25) + sqrt(Es/4)*filter(H{i,2},1,U26) +
sqrt(Es/4)*filter(H{i,3},1,U27) + sqrt(Es/4)*filter(H{i,4},1,U28),1,sig);
    end
end

% computing the single carrier estimates

```

```

if m == 1
    % 1x1 estimated signals
    for i=1:length(rho_tx)
        S_est1{i} = non_causal_filter(conj(H{i,1}),R1{i});
    end

elseif m == 2
    % 2x1 estimated signals
    for i=1:length(rho_tx)
        S_est1{i} = non_causal_filter(conj(H{i,1}),R1{i}) + filter( H{i,2},1,conj(R2{i}));
        S_est2{i} = non_causal_filter(conj(H{i,2}),R1{i}) + filter(-H{i,1},1,conj(R2{i}));
    end

elseif m == 3
    % 3x1 estimated signals
    for i=1:length(rho_tx)
        S_est1{i} = non_causal_filter(conj(H{i,1}),R1{i}) + filter( H{i,2},1,conj(R2{i})) +
        filter( H{i,3},1,conj(R3{i}));
        S_est2{i} = non_causal_filter(conj(H{i,2}),R1{i}) + filter(-H{i,1},1,conj(R2{i})) +
        filter( H{i,3},1,conj(R4{i}));
        S_est3{i} = non_causal_filter(conj(H{i,3}),R1{i}) + filter(-H{i,1},1,conj(R3{i})) + filter(-
        H{i,2},1,conj(R4{i}));
    end

elseif m == 4
    % 4x1 estimated signals
    for i=1:length(rho_tx)
        S_est1{i} = non_causal_filter(conj(H{i,1}),R1{i}) + filter( H{i,2},1,conj(R2{i})) +
        filter( H{i,3},1,conj(R3{i})) + filter( H{i,4},1,conj(R4{i}));
        S_est2{i} = non_causal_filter(conj(H{i,2}),R1{i}) + filter(-H{i,1},1,conj(R2{i})) +
        filter( H{i,3},1,conj(R5{i})) + filter( H{i,4},1,conj(R6{i}));
        S_est3{i} = non_causal_filter(conj(H{i,3}),R1{i}) + filter(-H{i,1},1,conj(R3{i})) + filter(-
        H{i,2},1,conj(R5{i})) + filter( H{i,4},1,conj(R7{i}));
        S_est4{i} = non_causal_filter(conj(H{i,4}),R1{i}) + filter(-H{i,1},1,conj(R4{i})) + filter(-
        H{i,2},1,conj(R6{i})) + filter(-H{i,3},1,conj(R7{i}));
    end
end

```

```
end
```

```
% Recollecting the estimates in a single sequence for demapping
```

```
if m == 1
```

```
    S_est{i} = [S_est1{i}];
```

```
elseif m == 2
```

```
    for i=1:length(rho_tx)
```

```
        S_est{i} = [S_est1{i} S_est2{i}];
```

```
    end
```

```
elseif m == 3
```

```
    for i=1:length(rho_tx)
```

```
        S_est{i} = [S_est1{i} S_est2{i} S_est3{i}];
```

```
    end
```

```
elseif m == 4
```

```
    for i=1:length(rho_tx)
```

```
        S_est{i} = [S_est1{i} S_est2{i} S_est3{i} S_est4{i}];
```

```
    end
```

```
end
```

```
% demapping, decoding and bit comparison for single carrier systems
```

```
for i=1:length(rho_tx)
```

```
    r_decimal{i} = pskdemod(S_est{i},M);
```

```
    r_bits{i} = mary_2_bin(log2(M),r_decimal{i});
```

```
    r_decoded{i} = vitdec(r_bits{i}, trel, 1, 'trunc', 'hard');
```

```
    number_of_bit_errors{i} = sum(xor(r_decoded{i}, x));
```

```
end
```

```
% summing up the number of bit errors for the given SNR value
```

```
bit_errors_1 = bit_errors_1 + number_of_bit_errors{1};
```

```
bit_errors_2 = bit_errors_2 + number_of_bit_errors{2};
```

```
bit_errors_3 = bit_errors_3 + number_of_bit_errors{3};
```

```
bit_errors_4 = bit_errors_4 + number_of_bit_errors{4};
```

```
bit_errors_5 = bit_errors_5 + number_of_bit_errors{5};
```

```
bit_errors_6 = bit_errors_6 + number_of_bit_errors{6};
```

```
bit_errors_7 = bit_errors_7 + number_of_bit_errors{7};
```

```
%    bit_errors_8 = bit_errors_8 + number_of_bit_errors{8};
```

```

end % bit iteration ends

% saving the number of bit errors for the iterated SNR value
total_bit_errors_1 = [total_bit_errors_1 bit_errors_1];
total_bit_errors_2 = [total_bit_errors_2 bit_errors_2];
total_bit_errors_3 = [total_bit_errors_3 bit_errors_3];
total_bit_errors_4 = [total_bit_errors_4 bit_errors_4];
total_bit_errors_5 = [total_bit_errors_5 bit_errors_5];
total_bit_errors_6 = [total_bit_errors_6 bit_errors_6];
total_bit_errors_7 = [total_bit_errors_7 bit_errors_7];
% total_bit_errors_8 = [total_bit_errors_8 bit_errors_8];

end % SNR iteration ends

% saving the number of bit errors for the iterated Monte Carlo run
total_bit_errors_monte_1 = [total_bit_errors_monte_1; total_bit_errors_1 ];
total_bit_errors_monte_2 = [total_bit_errors_monte_2; total_bit_errors_2 ];
total_bit_errors_monte_3 = [total_bit_errors_monte_3; total_bit_errors_3 ];
total_bit_errors_monte_4 = [total_bit_errors_monte_4; total_bit_errors_4 ];
total_bit_errors_monte_5 = [total_bit_errors_monte_5; total_bit_errors_5 ];
total_bit_errors_monte_6 = [total_bit_errors_monte_6; total_bit_errors_6 ];
total_bit_errors_monte_7 = [total_bit_errors_monte_7; total_bit_errors_7 ];
% total_bit_errors_monte_8 = [total_bit_errors_monte_8; total_bit_errors_8 ];

end % Monte Carlo iteration ends

% computing the bit error rates
bit_error_rate_1 = total_bit_errors_monte_1 / n;
bit_error_rate_final_1 = mean(bit_error_rate_1);

bit_error_rate_2 = total_bit_errors_monte_2 / n;
bit_error_rate_final_2 = mean(bit_error_rate_2);

bit_error_rate_3 = total_bit_errors_monte_3 / n;

```

```
bit_error_rate_final_3 = mean(bit_error_rate_3);
```

```
bit_error_rate_4 = total_bit_errors_monte_4 / n;  
bit_error_rate_final_4 = mean(bit_error_rate_4);
```

```
bit_error_rate_5 = total_bit_errors_monte_5 / n;  
bit_error_rate_final_5 = mean(bit_error_rate_5);
```

```
bit_error_rate_6 = total_bit_errors_monte_6 / n;  
bit_error_rate_final_6 = mean(bit_error_rate_6);
```

```
bit_error_rate_7 = total_bit_errors_monte_7 / n;  
bit_error_rate_final_7 = mean(bit_error_rate_7);
```

```
% bit_error_rate_8 = total_bit_errors_monte_8 / n;  
% bit_error_rate_final_8 = mean(bit_error_rate_8);
```

```
if MonteCarlo == 1
```

```
    figure ;  
    semilogy(EbNo_db,bit_error_rate_1,'xb-');grid on;  
    hold on; semilogy(EbNo_db,bit_error_rate_2,'xg-');  
    hold on; semilogy(EbNo_db,bit_error_rate_3,'xr-');  
    hold on; semilogy(EbNo_db,bit_error_rate_4,'xc-');  
    hold on; semilogy(EbNo_db,bit_error_rate_5,'xm-');  
    hold on; semilogy(EbNo_db,bit_error_rate_6,'xy-');  
    hold on; semilogy(EbNo_db,bit_error_rate_7,'xk-');  
    % hold on; semilogy(EbNo_db,bit_error_rate_8,'xb-');
```

```
else
```

```
    figure ;  
    semilogy(EbNo_db,bit_error_rate_final_1,'xb-');grid on;  
    hold on; semilogy(EbNo_db,bit_error_rate_final_2,'xg-');  
    hold on; semilogy(EbNo_db,bit_error_rate_final_3,'xr-');  
    hold on; semilogy(EbNo_db,bit_error_rate_final_4,'xc-');  
    hold on; semilogy(EbNo_db,bit_error_rate_final_5,'xm-');
```

```

hold on; semilogy(EbNo_db,bit_error_rate_final_6,'xy-');
hold on; semilogy(EbNo_db,bit_error_rate_final_7,'xk-');
% hold on; semilogy(EbNo_db,bit_error_rate_final_8,'xb-');
end

title('MISO scheme with STBC over SUI index = 2 and rho_tx =[0,0.1,0.2,0.4,0.6,0.8,1]')
xlabel('E_b/N_o');ylabel('BER');

legend('tx=0','tx=0.1','tx=0.2','tx=0.4','tx=0.6','tx=0.8','tx=1')

```

THIS PAGE INTENTIONALLY LEFT BLANK



## LIST OF REFERENCES

1. Y. Wang, Z. Zhou, and J. Wang, "Analysis of Spatial Fading Correlation for different Antenna Array," *IEEE Conference on Wireless Communications and Networking*, Vol. 2, pp.700-704, March 2004.
2. J. Li, and E. Y. Zhang, "Analysis of Spatial Correlation Properties of Wireless MIMO Channel," *IEEE Conference on Communications and Information Technology*, Vol. 1, pp.222-225, October 2005.
3. S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Select Area in Communications*, Vol. 16, No. 8, pp.1451-1458, October 1998.
4. V. Tarokh, H. Jafakhani, and A. R. Calderbank, "Space-Time Block Codes from Orthogonal Designs," *IEEE Transactions on Information Theory*, Vol. 45, No. 5, pp.1456-1467, July 1999.
5. B. Hassibi and B. M. Hochwald, "High-Rate Codes That Are Linear in Space and Time," *IEEE Transactions on Information Theory*, Vol. 48, No. 7, pp.1804-1824, July 2002.
6. E. Biglieri and G. Taricco, *Transmission and Reception with Multiple Antennas: Theoretical Foundations*, now Publishers Inc, Hanover, 2004.
7. C. K. Park and K. S. Min, "A Study on Spatial Correlation Characteristic of Array Antenna for Multi Antenna System," *IEEE Asia- Pacific Conference on Microwave*, Vol. 3, December 2005.
8. Institute of Electrical and Electronics Engineers, 802.16a, Air Interface for Fixed Broadband Wireless Access System-Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz, 1 April 2003, <http://ieeexplore.ieee.org>, last accessed 1 May 2006.
9. B. Vucetic and J. Yuan, *Space-Time Coding*, Wiley, England, 2003.

10. J. G. Proakis, *Digital Communications, Fourth Edition*, Mc-Graw Hill, New York, 2001.
11. H. D. Saglam, "Simulation Performance of Multiple-Input and Multiple-Output System Employing Single Carrier Modulation and Orthogonal Frequency Division Multiplexing," MSEE Thesis, Naval Postgraduate School, December 2004.
12. A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2003.
13. F. T. Ulaby, *Fundamentals of Applied Electromagnetics*, pp342-343, Prentice Hall, Pearson Education Inc, 2004.
14. M. K. Ozdemir, E. Arvas, and H. Arslan, "Dynamics of Spatial Correlation and Implications on MIMO Systems," *IEEE Magazine on Radio Communications*, Vol. 42, Issue 6, pp.S14-S19, June 2004.
15. P. Z. Peebles, Jr., *Probability, Random Variables, and Random Signal Principles*, Forth Edition, Mc-Graw Hill, New York, 2001.
16. S. Lin and D. J. Costello, *Error Control Coding*, Second Edition, Prentice Hall, New Jersey, 2004.

## **INITIAL DISTRIBUTION LIST**

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Chairman, Code EC  
Electrical and Computer Engineering Department  
Naval Postgraduate School  
Monterey, California
4. Murali Tummala, Code EC/Tu  
Electrical and Computer Engineering Department  
Naval Postgraduate School  
Monterey, California
5. Patrick Vincent  
Electrical and Computer Engineering Department  
Naval Postgraduate School  
Monterey, California
6. Nieh, Jo-Yen  
Chung Cheng Institute of Technology  
Tao-Yuan, Taiwan